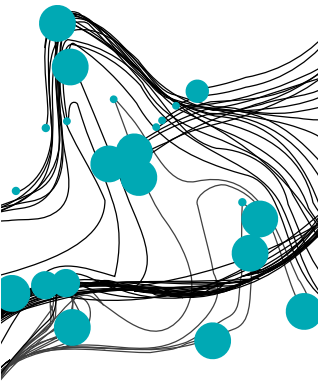
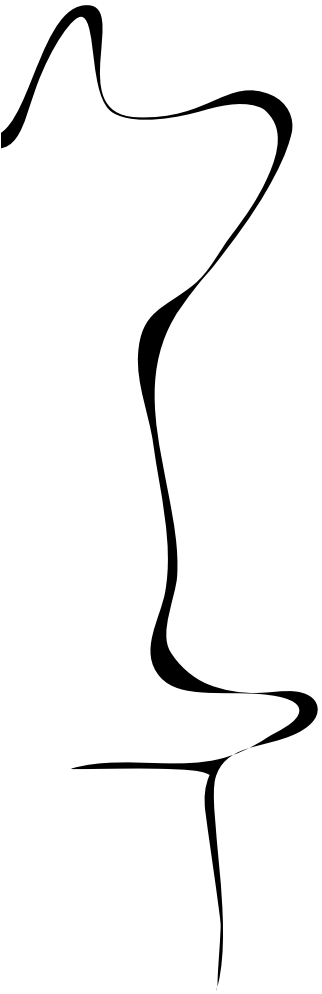


DMB

DATA MANAGEMENT AND BIOMETRICS



GROUPED PRODUCT RECOGNITION FROM IMAGES OF SUPERMARKET SHELVES USING MACHINE LEARNING

Hsin-Hui Huang

MASTER ASSIGNMENT

Committee:
dr. E. Talavera Martínez
dr.ing. Y. Huang

June, 2025

2025DMB0006
Data Management and Biometrics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

With the growing number and variety of products in supermarkets, managing retail shelves manually is time-consuming and prone to human error, making it difficult for staff to recognize and organize them efficiently. This work focuses on detecting and grouping similar or identical products from shelf images. In this project, we proposed a novel unsupervised, three-stage framework for grouped product recognition in supermarket environments, consisting of: (a) grocery product detection, (b) product characterization, and (c) grouped product recognition. For grocery product detection, we employ YOLOv5 to detect and locate each grocery object, and for product characterization, we extract multiple types of features, including CNN-based deep features, color histograms, shape and texture information, text from packaging, and product position on the shelf. Finally, in the grouped product recognition stage, we apply unsupervised clustering algorithms, including OPTICS and Agglomerative Clustering, to group similar products. We also evaluate the effectiveness of recent Vision Language Models (VLMs) for product detection and localization, and compare their performance with our proposed framework. Experimental results on public and real Dutch supermarket datasets show that the combination of CNN, color, and spatial features achieved the highest clustering performance, with an ARI of 0.7894, NMI of 0.8020, and Silhouette Score of 0.0358 on the Grocery Products dataset.

1 Introduction

The retail industry has undergone a significant transformation due to rapid technological evolution in recent years. The rise in living standards has led to an increased variety of retail products in supermarkets, making it challenging for both customers and employees to manage and recognize items manually [1]. In real retail settings, automatic detection and recognition of products displayed on the store shelf has provided the foundation of various advanced applications, which brings significant benefits to customers and businesses.

This study focuses on the recognition and grouping of similar or identical retail products on supermarket shelves. By grouping the same or similar products, it becomes possible to estimate the quantity of each product type. Grouped product recognition in supermarket settings can significantly help inventory management by enabling real-time stock tracking and automatic shelf monitoring from images of racks, which greatly reduces the staff workload [2]. Products on supermarket shelves are often arranged by brand or category. Grouped product recognition can also help check the display based on a planogram by monitoring shelf organization and ensuring correct placement. A *planogram* refers to a detailed diagram that shows how and where products should be arranged on shelves [3].

Real-world retail environments pose several challenges for object detection and recognition, including the detection of small objects obscured by complex backgrounds, the handling of cluttered and overlapping items on shelves, and the presence of products with arbitrary orientations [4]. Furthermore, visual similarities among products in terms of shape, color, and texture make accurate detection and recognition more difficult. Image-related issues such as different lighting conditions, blurring, and varying viewing angles can affect the reliability of detection and recognition models [5].

Computer vision plays a vital role in retail environments. Introducing computer vision-based object detection and recognition to improve customers' shopping experience, inventory management, store shelf layout, and process automation. For example, object detection systems in stores can automatically identify out-of-stock or misplaced products, enhancing operational efficiency. Object detection and recognition are foundational and challenging tasks in computer vision.

Object detection is the process of detecting the presence of different objects within an image or video frame [6]. It also forms the basis for advanced visual-language tasks such as object grounding [7], where the goal is to localize an object in an image based on a textual description, such as identifying "the red bottle on the top shelf." Object grounding introduces a language component, requiring the system to understand both the visual features of the scene and the semantic meaning of the query.

Early object grounding methods relied on CNN-based models, employing convolutional networks for image encoding combined with small-scale Long Short-Term Memory (LSTM) networks for language encoding. With the advancement of pre-trained and multimodal large models, state-of-the-art methods have significantly improved grounding performance. Recent state-of-the-art methods include Vision-Language Pre-trained (VLP) models such as Grounding DINO and Florence-2, which integrate visual detection with encoder-decoder architectures through cross-modal learning, as well as Multimodal Large Language Models (MLLMs) such as Qwen-VL and GPT-4o, which extend the capabilities of Large Language Models (LLMs) by integrating multiple data

types, such as images, textual information, audio, and more, and supporting cross-modal understanding through a unified architecture [7].

With the growth of deep learning, computer vision has made significant progress in recent years. However, object detection and recognition of retail products are still in their infancy, restricting the development of novel and advanced applications [8]. In the past few decades, researchers have been dedicated to addressing these issues to make progress in object detection. Recent improvements in object detection and recognition have been driven by deep learning technology such as Convolutional Neural Networks (CNNs). CNN-based methods such as Faster R-CNN, You Only Look Once (YOLO), and Single Shot MultiBox Detector (SSD) have been widely used. These CNN-based methods achieve impressive accuracy in various benchmarks when applied to object detection and classification [9].

In retail applications, studies show the effectiveness of deep learning methods in tasks such as automatic product recognition and checkout automation. While these achievements mark huge progress, many deep learning-based methods are still in their early stage [10]. One such underexplored area is grouped product recognition, which aims to recognize groups of similar or related products displayed together on shelves, instead of recognizing each product individually. This task introduces additional challenges, as many product classes display significant similarities in packaging design, making them harder for a single model to recognize or group them correctly [11].

This research proposes a novel framework to detect and group retail products from shelf images in supermarket environments by using machine learning techniques. The proposed framework is divided into three steps:

(1) **Grocery Product Detection** leverages You Only Look Once (YOLO) [12] to detect and localize items.

(2) **Product Characterization** extracts visual, spatial, and textual features.

(3) **Grouped Product Recognition** employs existing unsupervised clustering algorithms to group similar products together.

We evaluate this work using two public datasets, Grocery Products [13], WebMarket [14], as well as the Dutch Markets dataset, which we collected from Dutch supermarkets, to validate its effectiveness across different retail environments.

1.1 Research Questions

This study is guided by the following main research question:

- How to leverage visual features, textual information, and spatial relationships to recognize and group objects in retail environments in an unsupervised manner?

To support the investigation of the main research question, the following sub-questions were developed:

- What visual features, such as color and texture, are most effective for grouping retail products?
- How does textual information extracted from product packaging (e.g., labels, brand names) affect grouped product recognition?
- How do spatial relationships between products affect unsupervised grouping in supermarket environments?
- How effective are vision language models (VLMs) compared with the proposed framework in detecting and localizing products in dense retail environments?

In summary, the significant contributions of this work are as follows:

- A novel three-stage framework was proposed for grouped product recognition, consisting of grocery product detection (YOLOv5), product characterization, and clustering using OPTICS and Agglomerative algorithms. We also applied a per-image threshold optimization strategy based on Silhouette Score and entropy, which automatically selects the best clustering threshold for each image.
- This study conducted a comprehensive ablation study on feature combinations for grouped product recognition. Our pipeline integrated CNN-based visual features, color histograms, shape and texture descriptors, OCR-based textual features, and spatial layout. We evaluated the impact of each individual and combined feature on clustering performance.
- We present a new dataset, Dutch Markets, consisting of shelf images collected from Dutch supermarkets under real-world conditions. The dataset includes ground truth labels for product bounding

boxes and group assignments, enabling practical validation of grouped product recognition frameworks.

- Ground truth annotations were manually created for two public datasets (Grocery Products and WebMarket) to support the evaluation of clustering performance.

The organization of the paper is as follows: Section 2 introduces the literature review on retail product detection and recognition, and the proposed product recognition framework in Section 3. Section 4 provides details of the datasets and the experimental results. Section 5 presents a discussion of the experimental results. Finally, Section 6 draw a conclusion of the work.

2 Related Work

This section reviews previous studies on grocery product detection and recognition in retail settings. It first presents recent methods used for detecting and recognizing grocery items, followed by a discussion of the limitations in existing research that motivate this work.

2.1 Grocery Product Detection and Recognition

Many researchers have delved into the issues of object detection and recognition in videos and images. This section discusses the literal review of grocery product detection and recognition methodologies.

Tonioni et al. [15] developed a deep learning-based pipeline that uses YOLOv2 for object detection to obtain a set of bounding boxes as region proposals. The detection step identifies potential product regions without specifying the actual label. Candidate regions were then processed to generate global descriptors by using a pre-trained CNN embedder which backbone network is a VGG_16 [16].

To perform recognition, the global descriptor for each candidate region is matched against a pre-computed reference database created by similar descriptors from each available reference image. The comparison is done by calculating the similarity distance between query and reference descriptors through K-NN similarity search.

To address false detections and re-ranking potential matches, the refinement stage introduces a three-fold strategy. Since the initial ranking is based on global descriptors, the refinement step leverages local features extracted from both the query and reference images to better distinguish similar-looking products. The algorithm seeks similarity between these local descriptors to compute the matches. The results are then re-ranked based on the sum of weights between the local features.

An additional refinement step employs a distance ratio criterion [17] to filter out ambiguous recognitions. If the ratio between the query image and its 1-NN and 2-NN exceeds a defined threshold, the match is considered unreliable and discarded. Lastly, products in the same macro-category are often displayed together on the shelf. From the candidate regions, only those that have been identified with high confidence (0.1) by Detector are considered. For these high-confidence regions, this strategy uses 1-NN matches from high-confidence regions to deduce the macro-category. Accordingly, it then filters potential matches based on that category.

A two-stage pipeline model for object detection and recognition was proposed by Gothai et al. [18]. In the first step, the pre-processing step uses several filter methods, including Contrast Limited Adaptive Histogram Equalization (CLAHE), smoothing, and sharpening, to address lighting conditions, camera flash, and light reflection problems. The framework employs YOLOv5 for real-time object detection. YOLOv5 detects the products and generates bounding boxes along with their associated probabilities. Applying a Gaussian filter to remove false positives.

Product recognition uses logo images within the detected location. This stage relies on extracting and combining visual features such as color, shape, and size. For shape and size vocabulary, using Latent Dirichlet Allocation (LDA) to build a vocabulary of shape and size histogram features. Gaussian SIFT descriptors are used to extract local spatial features at multiple scales. Then, apply the Dirichlet function to transform histogram-based features, and BoW features are computed for each subregion.

In terms of color, images are converted to the HSV values, and then employ a 3D color picker to extract color features. With Naive Bayes, a color vocabulary is generated by clustering the color features. After that, Fisher Kernel Encoding is applied for color encoding and further refinement by leveraging Gaussian Mixture Model (GMM) to generate a visual word dictionary. Gradients are calculated based on how the feature vectors interact with the GMM components (weights, means, and variance). The gradients are aggregated into a Fisher vector, which is a Bag-of-Visual-words (BoV) extension. Fisher vector captures the probabilistic structure and the feature relationships. Concatenate the final vector by combining color, shape, and size features with a weighted strategy

and identify the brand by leveraging the final vector.

Selvam et al. [10] divided the product detection and recognition task into three modules:
(a) Retail product detection: Using YOLOv5 to create bounding boxes of products from video frames.
(b) Product-text detection: A curve-shaped text detection method is proposed. A Fully Convolutional Network (FCN) model with a ResNet50 backbone is introduced to extract features such as text regions (TRs), text center line (TCL), and its geometric attributes. ResNet50 includes skip connections to solve the vanishing gradient problem and enable the higher layer to perform as well as the lower layer. Applying masked TCL to retain only discrete text instances by removing background from TCL pixels.

After that, the enhanced striding algorithm is employed to refine the detection of text instances, accurately predicting text region geometries. It includes three steps: centralizing, striding, and sliding. After TCL generation, the bounding boxes are reconstructed around text instances with a post-processing technique named Width Height based Bounding Box Reconstruction (WHBBR). The algorithm transforms these irregular bounding boxes into accurate rectangular bounding boxes.
(c) Product-text recognition: This phase leverages a text recognition model, SCATTER, to recognize the text from the cropped text images [19]. This model consists of four steps that utilize CNN with a 29-layer ResNet backbone to extract features. Once the features are extracted, a Bi-LSTM is applied over the feature map to capture the relationships between characters in both directions. Lastly, CTC-Attention-based decoding is used to convert features into characters. Create the attention map from features and adapt a separate encoder-decoder to decode the attention map. Then, LSTM is used to obtain text characters.

Selvam et al. [20] introduced a three-stage grocery product recognition, combining (a) object detection, (b) OD-Refiner, and (c) object recognition.

Start with cutting-edge object detection named YOLOv5, generating bounding boxes around detected products. Despite YOLO being one of the most effective algorithms, YOLOv5 has difficulty detecting and locating small objects.

The second stage proposes a set of post-processing phases combined as "OD-Refiner". It addresses common challenges such as miss detection and inaccurate and overlapping bounding box predictions. YOLOv5 generates a number of redundant boxes that are irrelevant for accurate product detection. The Boolean Mask is applied to help retain only those boxes surpassing a probability threshold.

To further address overlapping problems, Non-max suppression (NMS) is employed. NMS evaluates bounding boxes based on Intersection over Union (IoU). If IoU between two bounding boxes exceeds a predefined threshold, the bounding box with the lower objectness score would be filtered, retaining only one bounding box.

However, NMS has limitations in dense and noisy situations. The Subtle-IoU layer is used to deal with this issue. The Subtle-IoU produces objectness scores for each bounding box through a fully convolutional layer. Then compute IoU between an estimated bounding box and an actual bounding box. Using a binary cross-entropy loss to learn a probabilistic score. Finally, the Subtle-IoU layer evaluates and creates singular detection for each product.

Furthermore, Missing box correction leverages planogram information, where the expected layout of items is compared against detected bounding boxes. By creating observed and reference planograms, it identifies similarities between layouts and corrects missed items by adjusting the detected frame.

Lastly, the object recognition phase adapts a DL-based optical character recognition system to process text from the product's packaging. Begin with a text detection that utilizes an Efficient Accurate Scene Text (EAST) detector [21] to identify and extract text regions from retail products. This model is a fully convolutional network that predicts text regions and their orientation angles.

Text recognition with a Batch Normalization Free Fully-Convolutional Rigorous Feature Flow Neural Network (BNFRNN) is applied to recognize text from detected text regions. BNFRNN comprises a fully convolutional network with two rigorous layers and a manuscript layer as its core. This model improves on previous methods by avoiding the use of batch normalization. Instead, this model uses a learnable scalar multiplier α. The Manuscript layer consists of the Taylor-SoftMax and CTC, which creates the final sequence. After that, the extracted text is compared with a predefined database, and the identity of the item is determined.

2.2 Research Gaps

Although many approaches have been developed for product detection and recognition, several challenges remain. Tonioni et al. [15] struggled with the slow test time speed, which is unsuitable for mobile or low-cost devices. Furthermore, the Detector and Embedder are developed separately, increasing the model's complexity. Also, they lacked sufficient product samples for training and had a domain gap between the training and testing images.

Selvam et al. [10] faced some challenges while their research achieved good performance when the product text was clearly visible. First, the framework completely relied on text from product packaging. Therefore, it failed if the text was occluded or missing. Moreover, the model was computationally expensive when training. The study did not cover real-world tasks such as tracking out-of-stock products, counting items, or detecting misplaced products on shelves.

A later study by Selvam et al. [20] explored a different strategy, introducing OD-Refiner to enhance detection performance. However, this framework still faced issues in recognizing partially visible or occluded text on packaging, which affected the accuracy of matching with the reference database. In addition, the framework was limited in recognizing specially designed or decorative characters, which are common on retail packaging.

Previous methods depend heavily on clear and complete textual information and lack the ability to estimate product quantities and monitor shelf inventory. To address these limitations, we propose a grouped product recognition framework that integrates visual appearance, textual information, and spatial relationships between products, instead of relying only on complete text. This framework also enables the system to count the visible number of items for each product type on the shelf, which is essential for inventory tracking and shelf monitoring.

3 Proposed Framework

The overall architecture of the proposed framework is illustrated in Figure 1. This framework comprises three stages: (i) Grocery Product Detection, (ii) Product Characterization, and (iii) Grouped Product Recognition. Each stage will be discussed in detail below.

3.1 Grocery Product Detection

YOLOv5 was adopted in our framework due to its effectiveness in recent state-of-the-art grocery product recognition studies. Especially, YOLOv5 is particularly effective in detecting densely packed and small objects.

YOLOv5 [22, 23, 24, 25, 26, 27] combines CSPDarknet53 and Path Aggregation Network (PANet), which contributes to faster model training and lower computation costs. It consists of three core components: a backbone network, a neck network, and a detection head.

YOLOv5 adopts CSPDarknet53 as its backbone. It incorporates Cross Stage Partial (CSP) network as part of the architecture, which tackles the issue of repetitive gradient information commonly found in larger backbones. CSPDarknet53 also utilizes the Focus layer to perform image slicing, replacing the first three layers of the YOLOv3 backbone. It helps reduce information loss while downsampling, lowers CUDA memory consumption, and accelerates training speed. By reducing the number of network parameters, the backbone decreases model complexity and improves inference speed.

After feature extraction by the backbone, the neck is applied to aggregate and refine feature representations. YOLOv5 introduces PANet as its neck network to boost information flow. PANet uses a new feature pyramid network (FPN) and pyramid attention network (PAN). FPN follows a top-down pathway that improves the semantic representation by propagating deep-layer semantics to the shallow layer through upsampling. PAN adopts a bottom-up pathway that propagates the localization information from the shallow layer to the deep layer, improving the localization capability at different scales. Thereby, PANet improves the localization accuracy.

The last component is the head. It produces three distinct outputs of feature maps at different scales to enable multi-scale object prediction, which helps the model detect different sizes of objects. In addition, YOLOv5 predicts bounding boxes as offsets relative to a set of predefined anchor boxes.

Given an input image, YOLOv5 predicts multiple bounding boxes. Each bounding box consists of five values: x, y, w, h, and confidence. The (x, y) coordinates denote the center of the box relative to the grid cell, while w and h are the box width and height relative to the entire image. The confidence score indicates the IoU between the predicted bounding box and the ground truth bounding box. We used YOLOv5 as a product detector to detect and localize individual grocery items within the shelf images. Since fine-grained classification is not required, each product in the training images was annotated with a single class label using a bounding box, simplifying the process and accelerating the training speed.

3.2 Product Characterization

In order to categorize detected products for grouped product recognition, we extracted a combination of visual, textual, and spatial features. Visual features include CNN-based learned representations that capture overall appearance, color histograms to describe overall color distributions, HOG descriptors for shape information, and texture patterns via Gabor and LBP filters, while textual features are detected and extracted from EAST and EasyOCR.

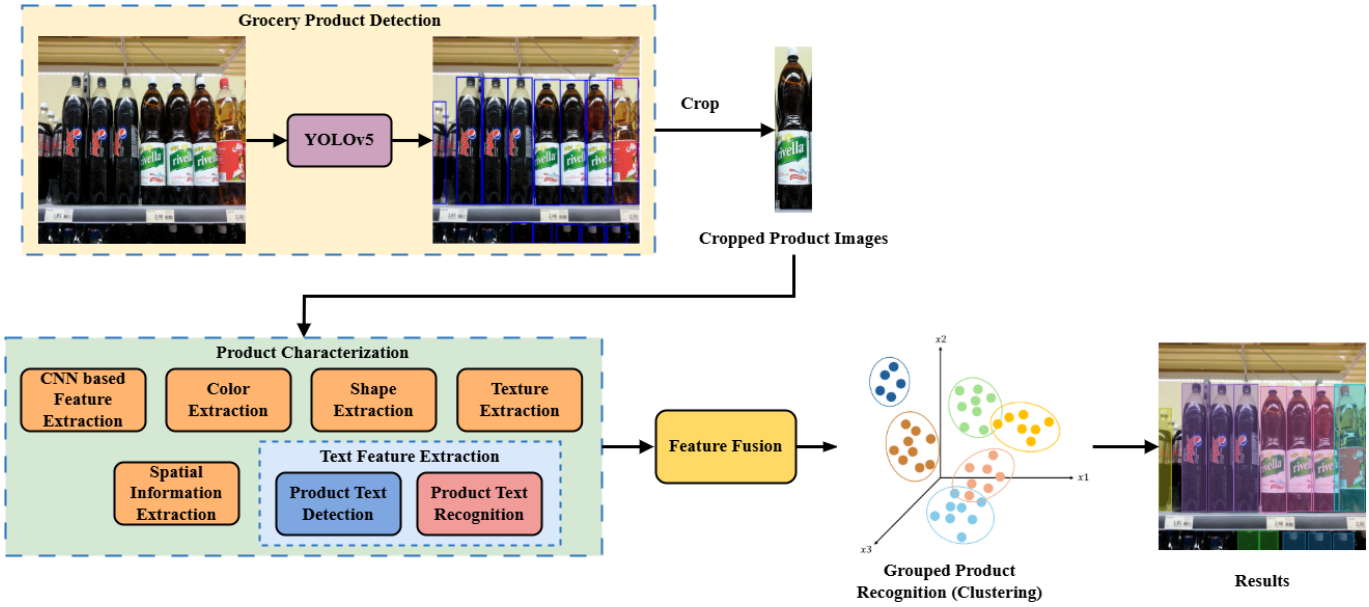


Figure 1: Pipeline architecture of the proposed grouped product recognition system.

Spatial information is also incorporated. In the following sections, we provide a detailed explanation of each feature extraction method.

CNN-based Feature Extraction

A Convolutional Neural Network (CNN) has demonstrated outstanding performance in image understanding tasks. Compared to traditional feature extraction models, CNN automatically captures diverse and important features, resulting in computational efficiency.

To leverage the robust feature extraction capabilities of deep learning, we adopted ResNet50 [28], a state-of-the-art CNN pre-trained on the ImageNet dataset. ResNet is a remarkable architecture that incorporates residual layers [29, 30, 31, 32, 33]. ResNet addresses the vanishing gradient issue by introducing shortcut connections, which offer an alternative shortcut for the gradient to pass more effectively through the network. The identity mapping is used to help avoid the over-fitting problem by enabling the model to bypass a CNN weight layer when the current layer is not beneficial. ResNet50 consists of 50 layers. The architecture of ResNet50 is shown in Figure 2.

Each cropped product image is resized to 224×224 pixels to match the input format of ResNet50. The processed image is then passed through the network to receive a 2048-dimensional feature vector for each product image by removing the last fully connected layer.

Color Feature Extraction

Color is one of the most fundamental features for differentiating retail products. Color histogram [34] is one of the most commonly adopted techniques for extracting color features from images. It captures the distribution of pixel intensities across predefined color bins. We utilized a color histogram in the HSV (Hue, Saturation, Value) color space to extract color features.

Although RGB is the most commonly used color representation in digital images, it does not closely align with human color perception. HSV color space is the closest to the human visual system. Hue indicates the type of color, which is defined as an angle from 0 to 360°. Saturation represents the purity of color, ranging from 0 to 1, and Value (Intensity) refers to the brightness of a color with a value in the range $[0, 1]$ [35]. HSV color histogram makes it more robust to lighting variations as it separates the luminance component (Value) from the chrominance components (Hue and Saturation) of a pixel's color [36, 37, 38].

The HSV color histogram effectively represents the overall color composition and brightness variations in product images. This is particularly useful for differentiating products that share similar forms but different color packaging.

Shape Feature Extraction

Shape features are extracted by using the Histogram of Oriented Gradients (HOG) [39], a widely used descriptor for capturing edge structures and local contours. HOG descriptor computes the accumulation of gradient direc-

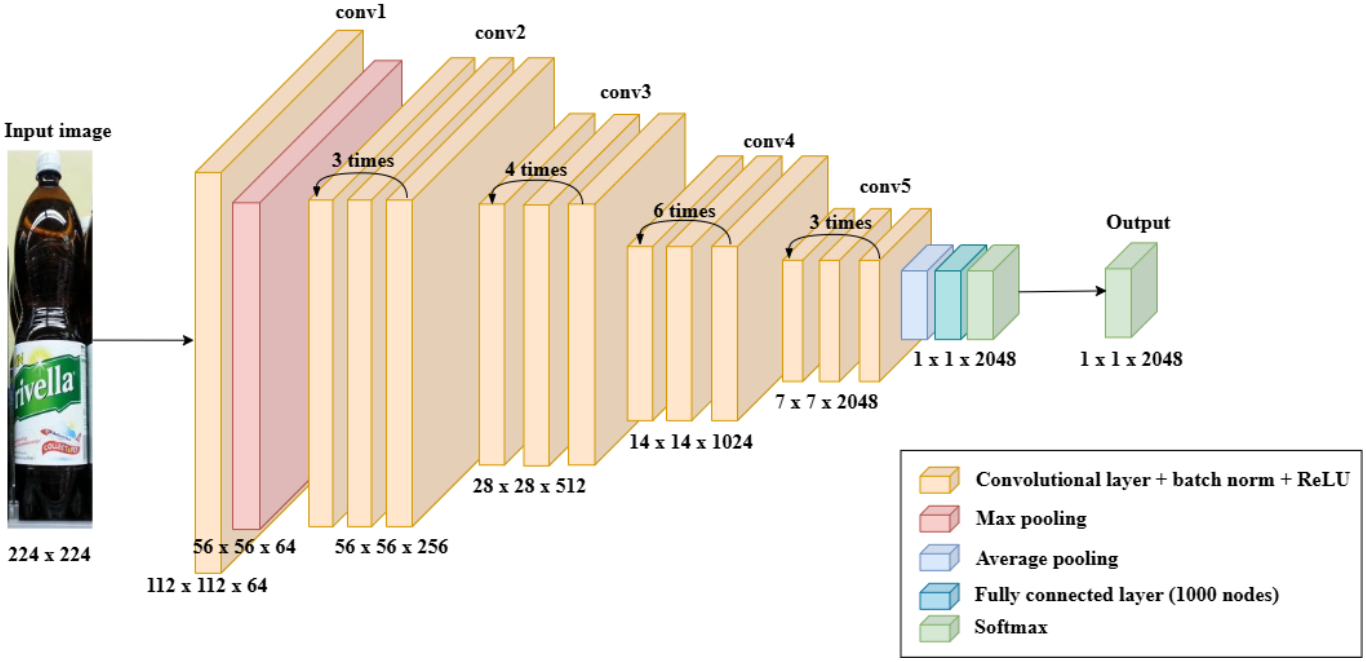


Figure 2: The architecture of ResNet50 for feature extraction, adapted from [28].

tions or edge orientations over the pixels within a small spatial region called "cell" of the image. Each image is first converted to grayscale to be well analyzed.

The image is divided into $N \times N$ small cells (e.g., 8×8 pixels). On each cell, calculate the gradient orientation shown in Equation 1 and the magnitude of each pixel based on Equation 2.

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{1}$$

$$M = \sqrt{G_x^2 + G_y^2} \tag{2}$$

A histogram is formed by accumulating the magnitudes of the same gradient over all pixels within the cell. Each pixel contributes to the histogram based on its orientation and magnitude. To account for lighting and contrast variations, neighboring cells are grouped into larger regions called "blocks" (e.g., 2×2 cells), and their histograms are normalized within each block [40, 41].

HOG descriptor are particularly beneficial for distinguishing between items with subtle packaging differences or unique shapes.

Texture Feature Extraction

We employed a combination of Gabor filters [42] and Local Binary Patterns (LBP) [43] to extract texture features, which are commonly used techniques in texture analysis.

Gabor filters [42] are bandpass filters that effectively capture texture patterns by analyzing their frequency content and orientation. An example of Gabor filter output is illustrated in Figure 3. A two-dimensional Gabor filter is defined as a Gaussian kernel function modulated by a sinusoidal wave:

$$G(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \cos\left(2\pi\frac{x'}{\lambda} + \psi\right) \tag{3}$$

where:

$$x' = x\cos\theta + y\sin\theta \tag{4}$$

$$y' = -x\sin\theta + y\cos\theta \tag{5}$$

Here, λ is the wavelength of the sinusoidal factor, θ is the orientation of the Gabor kernel, ψ is the phase offset, σ is the standard deviation of the Gaussian factor, and γ is the spatial aspect ratio that controls the ellipticity of the support of the Gabor function [44, 45].

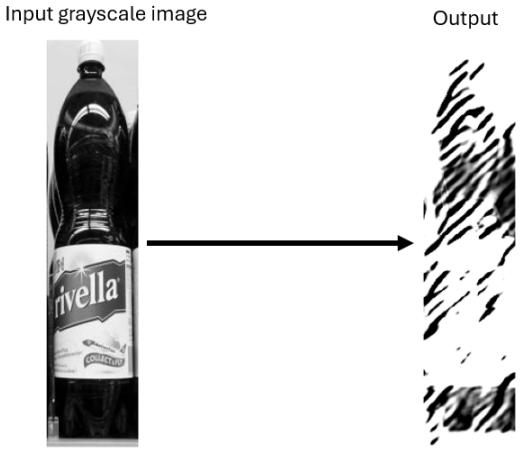


Figure 3: An example of applying a Gabor filter [42] to a grayscale image.

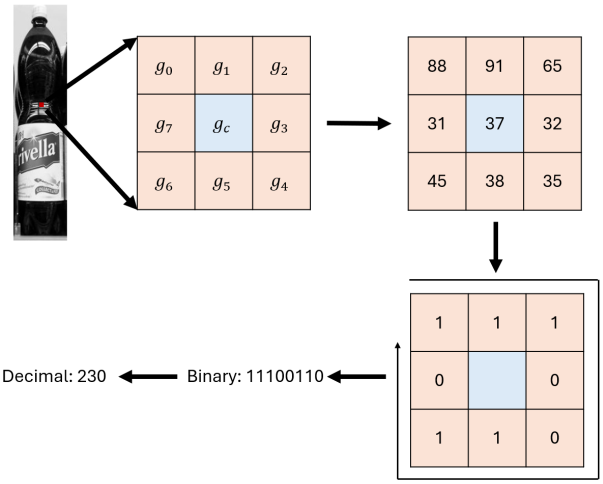


Figure 4: An illustration of the LBP [43] operator process.

Local Binary Pattern (LBP) [43] is a powerful texture operator that was first introduced by Ojala et al. Each pixel in the image is treated as a center point, and its intensity is compared with the intensities of its neighboring pixels within a specified radius. If a neighboring pixel has a higher or equal intensity, it is assigned a value of 1; otherwise, it is assigned to 0. The binary result is then converted into a decimal value, demonstrated in Figure 4. By applying this process to every pixel in the image, a set of LBP codes is generated and summarized into a histogram [46, 47, 44]. The LBP result is computed as:

$$LBP_{P,R}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \tag{6}$$

where the function $s(x)$ is defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Here, g_c denotes the intensity of the center pixel, and P is the number of neighbors surrounding the center point within a radius R.

Spatial Feature Extraction

In addition to visual features, spatial information is crucial for distinguishing grouped product categories, particularly on dense retail shelves.

To obtain this information, we implemented it by parsing bounding box values generated by YOLO product detector. The value of each detected object includes normalized coordinates for the center of the box (x, y), the width w, and the height h. For each detected object, a five-dimensional spatial feature vector is constructed: $[x, y, w, h, d]$, where d indicates the Euclidean distance of the object center to the image center, calculated based on Equation 8.

$$d = \sqrt{(x - 0.5)^2 + (y - 0.5)^2} \tag{8}$$

Text Feature Extraction

To extract meaningful textual features from product packaging, we incorporated the Efficient and Accurate Scene Text (EAST) Model [21] for Text Detection and EasyOCR [48] for Text Recognition, as illustrated in Figure 5.

Product Text Detection

We adopted the EAST detector [21], a fully convolutional network (FCN) that predicts word-level text regions in arbitrary orientations and quadrilateral shapes directly from images.

The model produces two key outputs: a score map and a geometry map. The score map denotes the confidence score of predicted text regions, and the geometry map describes the shape and spatial structure of the bounding boxes around the text regions. The geometry map can be represented in two formats: RBOX, which includes distances to box boundaries and a rotation angle, or QUAD, which consists of coordinate offsets to the four vertices of the quadrangle.

Non-Maximum Suppression (NMS) is applied as a post-processing step to suppress overlapping boxes to re-

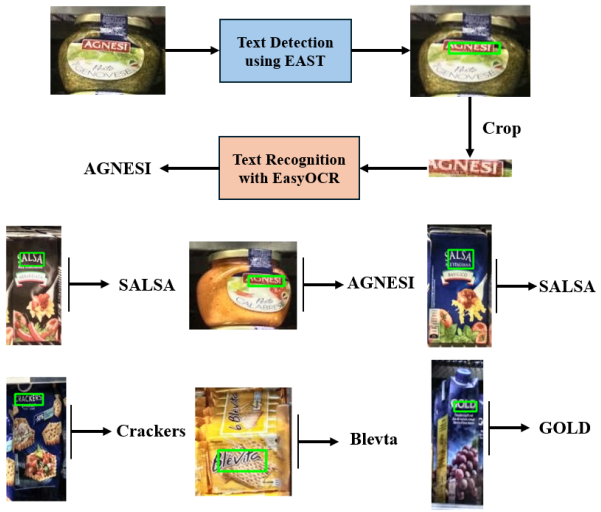


Figure 5: Text detection and recognition pipeline using EAST [21] and EasyOCR [48], with examples of extracted text.

ceive the final output.

Product Text Recognition

After Text Detection, the cropped text regions are then passed to Product Text Recognition through EasyOCR [48], a Python-based library built on PyTorch that provides robust and accurate text recognition. EasyOCR integrates convolutional feature extraction with a Long Short-Term Memory (LSTM) sequence labeling model and Connectionist Temporal Classification (CTC) decoding. It supports over 80 languages, including English, Arabic, Chinese, and Latin, making it well-suited for multilingual product packaging.

Feature Fusion

Different types of features are combined together in the feature fusion stage. These include CNN-based visual features, color, shape, texture features, and the spatial information of the products on the shelf. Text information is extracted using EasyOCR and vectorized using Term Frequency–Inverse Document Frequency (TF-IDF) [49]. TF-IDF is a commonly used technique in information retrieval and natural language processing to measure how important a word is in a document relative to a corpus.

TF-IDF score for a term t in a document d is calculated as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

where:

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$\text{IDF}(t) = \log\left(\frac{N}{1 + n_t}\right)$$

Here, $f_{t,d}$ is the frequency of term t in document d, $\sum_{t' \in d} f_{t',d}$ is the total number of terms in document d, N is the total number of documents in the corpus, and n_t is the number of documents that contain the term t.

All extracted features were concatenated into a single feature vector. This allows the clustering algorithm to make better decisions when grouping products. An overview of the extracted features and their corresponding dimensions is presented in Table 1.

Extracted Feature	Dimensions
CNN-based features	2048
Color features (HSV)	512
Shape features (HOG)	7200
Texture features (Gabor filters + LBP)	257
Spatial features	5
Text features	50

Table 1: Summary of feature dimensions.

3.3 Grouped Product Recognition

To identify groups of similar products on the shelves, we applied an unsupervised clustering strategy based on extracted features. For the clustering process, we employed two algorithms: OPTICS (Ordering Points To Identify the Clustering Structure [50] and Agglomerative Clustering [51].

OPTICS

OPTICS [50] is a density-based clustering method that extends the principles of DBSCAN by capturing the clustering structure of data points across multiple density levels. Unlike DBSCAN, which requires a fixed neighborhood radius ϵ to detect clusters, OPTICS generates a reachability plot that reflects how data points cluster across a range of density thresholds, without requiring a fixed ϵ value.

OPTICS introduces two core concepts: core distance and reachability distance. The core distance of a point p is defined as the smallest distance needed to reach a certain number of neighboring points, determined by the parameter called $MinPts$. To find this distance, calculate

how far p is from its $MinPts$-th neighbor. If the point does not have enough neighbors, it is not considered a core point. The reachability distance between the core point p and a point o within the radius ϵ is calculated by comparing the core distance of p and the distance between p and o, and taking the larger of the two.

OPTICS stores both the core distance and the reachability distance for each point and produces a total order of points based on reachability. These values are used to construct the reachability plot, a visualization that helps identify clusters of varying densities. In this plot, lower reachability values indicate dense regions, while higher values imply sparse areas or noise.

OPTICS algorithm relies on several hyperparameters that influence the resulting clustering structure. The *min_samples* parameter defines the minimum number of neighbors required for a point to be considered a core point. The *xi* parameter determines the steepness in the reachability plot required to define a cluster boundary. Additionally, *min_cluster_size* defines the minimum number of points in a cluster, which is optional. The hyperparameter settings used in our implementation are listed in Table 2.

Hyperparameter	Value
min_samples	2
xi	0.01 - 0.10
min_cluster_size	None

Table 2: Hyperparameters used for OPTICS [50] and their selected value in our implementation.

Agglomerative Clustering

Agglomerative Clustering [51] is a bottom-up hierarchical clustering method. It starts by considering each data point as its own cluster and then gradually merging two clusters with minimal distance. This merging repeats until all points are grouped into a single large cluster or a stopping condition is met.

In our implementation, *n_clusters* defines the number of clusters that can be generated. *distance_threshold* determines the maximum distance at which points can be merged into the same cluster. Lastly, *linkage* criterion controls how the distance between clusters is computed. We use the ward method, which minimizes the variance within clusters. The values used in our implementation are summarized in Table 3.

Hyperparameter	Value
n_clusters	None
distance_threshold	20 - 200
linkage	ward

Table 3: Hyperparameters used for Agglomerative Clustering [51] and their selected value in our implementation.

Threshold Optimization

In addition to using a fixed threshold for clustering, this study explored two dynamic threshold optimization strategies: one based on the Silhouette Score and the other based on entropy. These strategies are applied to the distance_threshold in Agglomerative Clustering and the xi parameter in OPTICS, both of which control how clusters are formed.

For the Silhouette Score-based method, a range of candidate thresholds is defined for each image. For each threshold, we apply the clustering algorithm to the image. After clustering, we compute the Silhouette Score for each clustering result. Once all thresholds have been evaluated, we select the threshold that achieves the highest Silhouette Score as the optimal threshold. Figure 6 illustrates an example of this process, demonstrating how the Silhouette Score varies with the threshold and how the optimal value is selected.

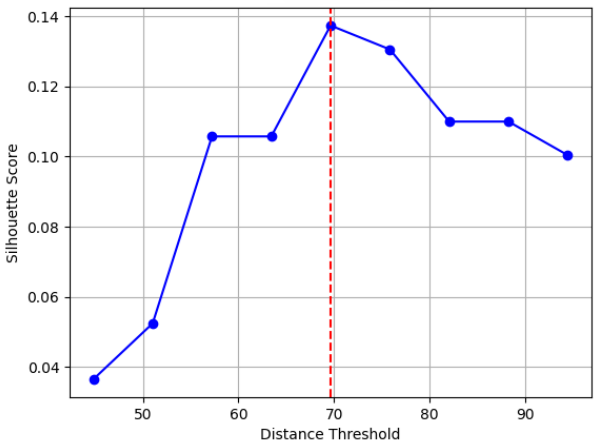


Figure 6: An example of Silhouette Score plotted against different distance thresholds under Agglomerative Clustering [51] for a single image. The red dashed vertical line indicates the optimal threshold that achieves the highest Silhouette Score.

For the entropy-based method, the optimal threshold is determined by calculating the entropy of the resulting cluster distribution. Entropy measures the level of disor-

der in the grouping. Lower entropy indicates more concentrated clusters, while higher entropy suggests more disorderly or ambiguous grouping. The optimal threshold is selected from a range of candidate thresholds as the one that results in the lowest entropy. Shannon's entropy is defined as follows [52]:

$$H(x) = -\sum_{x} p(x) \log_2 p(x)$$

where $p(x)$ denotes the probability of samples assigned to cluster x.

Figure 7 demonstrates how the entropy of the clustering result varies across different distance threshold values. The lowest entropy value corresponds to the selected optimal threshold.

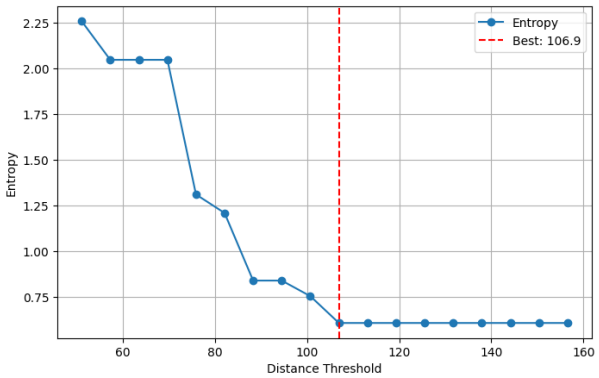


Figure 7: An example of entropy [52] plotted against different distance thresholds under Agglomerative Clustering [51] for a single image. The red dashed vertical line indicates the optimal threshold that achieves the lowest entropy.

3.4 Object Grounding through Vision Language Models

To explore the potential of Vision Language Models (VLMs) in object detection, we adopted two methods: InternVL2.5 [53] and GPT-4o [54]. InternVL2.5 [53] is a recent Multimodal Large Language Model (MLLM). InternVL2.5 supports various vision-language tasks, including image captioning and object grounding. It performs object grounding by understanding both visual and textual input and outputs bounding boxes around regions in the image.

GPT-4o [54] is a state-of-the-art Multimodal Large Language Model (MLLM) developed by OpenAI. It is capable of multimodal inputs, including text, image, or audio, and generates outputs across several modalities. GPT-4o adopts an end-to-end architecture across all modalities processed by a unified neural network. It features fast response times that are comparable to human-level interaction.

We evaluate InternVL2.5 and GPT-4o on benchmark datasets. For each image, the models are given different prompts to generate bounding boxes for individual products. The prompts used are provided in appendix A.

4 Experimental Results

4.1 Datasets

For our experimental evaluation, we use two publicly available datasets: Grocery Products [13] and WebMarket [14], as well as the Dutch Markets dataset, as can be seen in Figure 8.

Although the datasets include existing annotations, we noticed that many of the bounding boxes were not precise. To improve annotation consistency and accuracy, we manually refined the bounding boxes in both the WebMarket and Grocery Products datasets by correcting inaccurate boxes and adding missing ones. The annotation process was conducted using Roboflow, a computer vision platform that supports data annotation, model training, and deployment.

Each product in the images was manually annotated with bounding boxes. Since the goal was to detect where products appear in the image, rather than to classify them. Therefore, all annotations were assigned the same class label of 0. In addition, to evaluate the performance of our grouped product recognition method, we manually created ground truth labels by assigning the same label to products belonging to the same group.

Grocery Products [13] is used for fine-grained object classification and localization. It contains hierarchical categorical information for every product. It has a total of 80 product categories. However, only 27 of those 80 classes contain ground truth. Therefore, only those 27 product categories would be used in the recognition and localization task, including 3235 fine-grained product pictures. The testing set includes 680 shelf images, collected with a mobile phone under various lighting settings, camera angles, and zoom scales in real retail environments. Each image contains between 6 and 30 products.

WebMarket [14] consists of 3153 shelf images of size 2272 × 1704 with 102 product categories. The images were taken from 10 different grocery shelves, and each image covers three to four shelves. Most of the pictures were shot from the front without using any special lighting. The ground truth only provides information

Dataset	Given labels	Provided labeling	Total images	Training set	Validation set	Testing set
Grocery Products[13]	Class label (e.g., Food/Cereals) & bounding box	Class label (0 / Group ID) & bounding box	680	544	102	34
WebMarket[14]	Class label (object) & bounding box	Class label (0 / Group ID) & bounding box	300	240	45	15
Dutch Markets	-	Class label (0 / Group ID) & bounding box	128	-	-	128

Table 4: Overview of the datasets used in this study, including original labels, provided annotations, and the number of images in each data split.

about which shelf image each product appears in, without precise location annotations. Therefore, the location of the products must be manually identified.

In addition to the public datasets, we collected 128 shelf images from Dutch supermarkets, which we refer to as the ***Dutch Markets*** dataset. The images were captured using a phone under natural in-store lighting conditions. Each image contains around 5 to 40 visible product items. These images are used to evaluate model performance on real-world retail environments, as illustrated in Figure 8. To enable evaluation, we manually annotated all products in the images with bounding boxes using Roboflow, following the same annotation approach used for the public datasets.

In this study, the Grocery Products and WebMarket datasets were divided into a 85% training set, a 15% validation set, and a 5% testing set. The Dutch Markets dataset, manually annotated for evaluation, was used for testing, as summarized in Table 4.

(a) Samples of images in the Grocery Products dataset [13]

(b) Samples of images in the WebMarket dataset [14]

(c) Samples of images in the Dutch Markets dataset

Figure 8: Example shelf images from two public datasets and a self-collected dataset from Dutch supermarkets.

4.2 Evaluation Metrics

In this work, distinct evaluation metrics are applied at different stages of our framework. Product detection stage is evaluated by Precision, Recall, and F1-score. For the grouped product recognition stage, we employ clustering evaluation metrics, including Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Silhouette Score, to measure the quality of product grouping.

Evaluation metrics of Product Detection

The performance of product detection was assessed based on three evaluation metrics: Precision, Recall, and F1-score.

- **Precision**: Measures the proportion of correctly detected products among all detected products. It reflects how accurate the model's detections are, calculated based on Equation 9.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

where TP is the number of correct detections and FP is the number of incorrect detections.

- **Recall**: It measures the proportion of correctly detected products among all actual products present in the images. It indicates the model's ability to detect all relevant objects, evaluated by Equation 10.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

where FN is the number of missed detections.

- **F1-score**: Represents the harmonic mean of Precision and Recall. It helps measure both the model's accuracy and its ability to find all products, ranging from 0 to 1. It is computed as:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

Evaluation metric of object grounding

Accuracy is used as the evaluation metric for object grounding, same as [55]. It is defined as the ratio of correctly grounded object bounding boxes to the total number of predicted grounded boxes. A predicted box is considered correct if its Intersection over Union (IoU) with the corresponding ground truth box exceeds a threshold of 0.5.

Evaluation metrics of Grouped Product Recognition

For evaluating the quality of grouped product recognition, we employ three clustering evaluation metrics: Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Silhouette Score, capturing both external and internal validation aspects.

- **Adjusted Rand Index (ARI)**: It measures the similarity between predicted and true groupings by checking all possible pairs of items, adjusted for random chance. A score of 1 means perfect matching, while a score close to 0 indicates random assignment. It is calculated as:

$$\text{ARI} = \frac{RI - Expected[RI]}{\max(RI) - Expected[RI]} \tag{12}$$

where RI is the Rand Index and $Expected[RI]$ is its expected value of RI if clusters were assigned randomly. $\max(RI)$ is the maximum possible value of RI. ARI ranges from -1 to 1.

- **Normalized Mutual Information (NMI)**: It quantifies the shared information between the predicted and true labels, normalized to scale between 0 and 1. Higher scores indicate stronger agreement and better clustering quality, measured by Equation 13.

$$\text{NMI}(U, V) = \frac{2 \times I(U; V)}{H(U) + H(V)} \tag{13}$$

where $I(U; V)$ is the mutual information between ground truth clusters U and predicted clusters V assigned by the algorithm, and $H(\cdot)$ implies entropy.

- **Silhouette Score**: It is an internal measure of clustering quality. It tells how similar a sample is to its own cluster compared to other clusters. The score ranges from -1 to 1, where a higher value indicates that the sample is well-matched to its own cluster and poorly matched to neighboring clusters. The formula is shown in Equation 14.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{14}$$

where $a(i)$ is the average distance between sample i data point and all other points in the same cluster, and $b(i)$ is the average distance between sample i data point and all points in the nearest cluster.

4.3 Grocery Product Detection Performance

As we can observe in Table 5, our implementation based on YOLOv5 achieved a precision of 89.0%, a recall of 89.2%, and an F1-score of 89.1% on the Grocery Products dataset. On the WebMarket dataset, the method achieved a precision of 92.1%, a recall of 93.8%, and an F1-score of 92.94%.

In addition, we evaluated the product detection performance of our implementation by comparing it with several existing studies, as well as the best-performing object grounding results obtained by GPT-4o and InternVL2.5 (from Table 7), on the Grocery Products and WebMarket datasets, as summarized in Table 5.

On the Grocery Products dataset, various detection methods based on different technical frameworks were compared. Early works based on traditional feature-based methods, such as Bag of Words (BOW) and deep learning models like Deep Neural Network (DNN) [56], achieved relatively lower precision and recall. [57] and [58] introduced CNN-based models, showing improvement in F1-score compared to traditional methods. The YOLOv5-based method by [10] achieved the highest precision (92.1%) but with lower recall (86.8%) compared to our implementation. In contrast, we enhanced YOLOv5's detection capabilities, achieving a higher F1-score of 89.1%. Additionally, the grounding-based model InternVL2.5 reported significantly lower performance, with a precision of 15.0%, a recall of 7.68%, and an F1-score of only 10.16.

On the WebMarket dataset, the detection performance of our implementation was compared against several existing approaches. Earlier approaches, such as ERP+CNN and R-CNN-G, achieved lower F1-score, which were below 80%. With the adoption of YOLOv5

by [10], detection accuracy improved significantly, achieving an F1-score of 86.3%. [20] further improved the YOLOv5 performance by introducing a post-processing technique called OD-Refiner, achieving the highest reported precision of 92.56%, but their recall was comparatively lower at 85.64%, resulting in an F1-score of 88.97%. By contrast, our YOLOv5-based implementation achieved a more balanced detection, with a precision of 92.1%, a recall of 93.8%, and an F1-score of 92.94%, the highest recall and F1-score among all compared methods. Furthermore, the performance of the grounding-based model GPT-4o was considerably low, with an F1-score of 6.1%.

Method	Precision (%)	Recall (%)	F1-score (%)
Grocery Products [13]			
[56] (BOW)	77.7	76.5	-
[56] (DNN)	73.1	73.6	-
[58] (ERP+CNN)	-	-	81.05
[57] (R-CNN-G)	-	-	80.21
[10] (YOLOv5)	**92.1**	86.8	83.3
Our implementation (YOLOv5)	89.0	**89.2**	**89.1**
Our implementation (InternVL2.5, grounding)	15.0	7.68	10.16
WebMarket [14]			
[58] (ERP+CNN)	-	-	78.76
[57] (R-CNN-G)	-	-	75.50
[10] (YOLOv5)	89.4	88.2	86.3
[20] (YOLOv5 + OD-Refiner)	**92.56**	85.64	88.97
Our implementation (YOLOv5)	92.1	**93.8**	**92.94**
Our implementation (GPT-4o, grounding)	12.8	4.0	6.1

Table 5: Comparison of product detection performance with existing methods and the best-performing object grounding result (from Table 7) on the Grocery Products [13] and WebMarket [14] datasets.

In addition to two public datasets, we evaluated both YOLOv5 models, one trained on the Grocery Products dataset and the other on WebMarket, using the Dutch Markets dataset. We also included the best-performing object grounding result for comparison. As shown in Table 6, the model trained on Grocery Products performed better, with a precision of 91.9%, a recall of 89.9%, and an F1-score of 90.9%. In comparison, the WebMarket-trained model had weaker performance, with an F1-score of 82%. In contrast, the grounding-based model GPT-4o demonstrated significantly lower performance, with an F1-score of only 6.01%.

Figure 9 demonstrates sample detection results on the Grocery Products, WebMarket datasets, and Dutch Markets dataset. Our implementation successfully captures products across various shelf layouts and effectively handles small, densely packed items.

Model	Precision (%)	Recall (%)	F1-score (%)
YOLOv5 - Grocery Prodcuts	**91.9**	**89.9**	**90.9**
YOLOv5 - WebMarket	80.7	83.3	82.0
GPT-4o (grounding)	9.18	4.47	6.01

Table 6: Performance comparison of two YOLOv5 models and the best-performing object grounding result (from Table 7) on the Dutch Markets dataset.

4.4 Object Grounding Performance through InternVL2.5 and GPT-4o

To evaluate the object grounding capability using InternVL2.5 [53] and GPT-4o [54], we designed different prompts to help the models in locating items by returning bounding box predictions.

Dataset	InternVL2.5			GPT-4o		
	P1	P2	P3	P4	P5	P6
Grocery Products [13]	0.29	15.00	0.62	2.45	9.73	14.45
WebMarket [14]	0.19	6.45	2.77	9.94	6.91	12.83
Dutch Markets	0.40	7.50	5.00	9.18	8.77	6.22

Table 7: Comparison of object grounding performance on Grocery Products [13], WebMarket [14], and Dutch Markets datasets using InternVL2.5 [53] and GPT-4o [54], with prompts detailed in appendix A.

Table 7 demonstrates a comparative evaluation for object grounding across the Grocery Products, WebMarket, and Dutch Markets datasets. InternVL2.5 achieved its highest accuracy with prompt 2 (15%), while GPT-4o generally outperformed InternVL2.5, reaching up to 14.45% accuracy on Grocery Products with Prompt 6. Similar to the WebMarket and Dutch Markets datasets, GPT-4o again outperformed InternVL2.5, demonstrated more stable and higher average performance. Prompt 6 achieved the best performance for GPT-4o on the WebMarket dataset, with an accuracy of 12.83%. For the Dutch Markets dataset, prompt 5 resulted in the highest accuracy of 9.18%. These findings indicate that GPT-4o demonstrates more stable and robust performance across different prompts and datasets.

(a) Grocery Products dataset

(b) WebMarket dataset

(c) Dutch Markets dataset

Figure 9: Prediction results of product detection on Grocery Products [13], WebMarket [14], and Dutch Markets datasets. Green boxes indicate correct detections (true positives), yellow boxes represent false detections (false positives), and red boxes highlight ground truth objects that were missed by the model (false negatives).

(a) Prompt 1 (InternVL) (d) Prompt 4 (GPT-4o)

(b) Prompt 2 (InternVL) (e) Prompt 5 (GPT-4o)

(c) Prompt 3 (InternVL) (f) Prompt 6 (GPT-4o)

Figure 10: Comparison of object grounding results using six different prompts. The left column shows results from InternVL2.5 [53] (Prompts 1–3), while the right column shows results from GPT-4o [54] (Prompts 4–6). Red and green bounding boxes indicate the detected product regions by InternVL2.5 [53] and GPT-4o [54], respectively.

Figure 10 illustrates the visual results of object grounding from six different prompts across InternVL2.5 and GPT-4o. Both models demonstrated limited success in accurately detecting individual products. InternVL2.5 (Prompt 1-3) generated sparse and incorrect results across prompts, while GPT-4o (Prompt 4-6) exhibited more consistent coverage but often failed to distinguish product boundaries. Many products were either missed or inaccurately localized by both models. These results show that current VLMs still struggle to detect and localize individual products in crowded and complex shelf environments.

4.5 Grouped Product Recognition Performance

Since clustering algorithms assign arbitrary cluster labels, a remapping process was necessary to align predictions with ground truth labels. Each cropped detection was matched to a ground truth object based on an IoU threshold. Cropped images were then assigned the corresponding ground truth label. For each image, the predicted clusters were mapped to ground truth classes based on majority voting within the cluster.

Table 8 shows the performances across different feature combinations on the Grocery Products dataset. When using the single feature, CNN achieved the best results in ARI and NMI under both Agglomerative Clustering and OPTICS, outperforming traditional visual features such as color, shape, texture, and spatial layout. When combining features, the performance improved significantly. The combination of CNN + Color + Spatial achieved the highest ARI (0.7894) and NMI (0.8020) under Agglomerative Clustering, and also attained a relatively higher Silhouette Score (0.0385), followed by the CNN + Color which reached an ARI of 0.7894, an NMI of 0.7989, and a Silhouette Score of 0.0359.

Under OPTICS clustering on the Grocery Products dataset. Among individual features, Color and CNN achieved the best performance, with Color obtaining an ARI of 0.6176 and NMI of 0.6741, and CNN achieving an ARI of 0.6162 and NMI of 0.6715. The combination of CNN + Color + Texture surpassed other combinations, with ARI and NMI reaching 0.7318 and 0.7597. The second-best performance is CNN + Color + Spatial, which reported an ARI of 0.7085 and an NMI of 0.7513. Compared to Agglomerative Clustering, OPTICS shows lower performance.

Table 9 demonstrates the performance on the WebMarket dataset under various feature combinations. In Agglomerative Clustering, CNN outperformed other single features, achieving the highest ARI (0.6582) and NMI (0.8014) scores. Combining CNN with Color and Texture achieved even better results, reaching the highest ARI (0.6858) and NMI (0.8183) across all combinations. Under OPTICS, clustering scores were generally lower compared to Agglomerative Clustering. CNN again achieved the best performance among single features (ARI = 0.4498, NMI = 0.7479). When combining different features, CNN + Color + Spatial produced the best performance with an ARI of 0.4713 and an NMI of 0.7683. CNN + Color + Texture was slightly lower, which achieved an ARI of 0.4520 and an NMI of 0.7652.

Spatial relationships play an important role in

Table 8: Performance on the Grocery Products dataset [13] using different feature combinations

Feature	Agglomerative Clustering			OPTICS		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.7303	0.7485	0.0158	0.6176	0.6741	-0.0040
Color	0.6207	0.6664	0.0688	0.6162	0.6715	-0.0347
Shape	0.5983	0.6473	0.0050	0.4171	0.5313	-0.0317
Texture	0.1025	0.1750	0.0496	0.0812	0.1540	0.0948
Spatial	0.0913	0.1647	-1.000	0.5462	0.5854	-0.1431
Text	0.1055	0.1720	0.8787	0.1185	0.1830	0.7819
CNN + Color	0.7894	0.7989	0.0359	0.6976	0.7460	-0.0187
CNN + Shape	0.7066	0.7240	-0.0034	0.5280	0.6052	-0.0257
CNN + Texture	0.7229	0.7540	0.0144	0.6335	0.6911	-0.0015
CNN + Spatial	0.7303	0.7583	0.0166	0.6082	0.6771	-0.0016
CNN + Text	0.7270	0.7421	0.0144	0.6224	0.6782	-0.0070
Color + Shape	0.6835	0.7298	-0.0028	0.5371	0.6284	-0.0032
Color + Texture	0.6426	0.6762	0.0694	0.5747	0.6396	-0.0288
Color + Spatial	0.6304	0.6779	0.0681	0.6248	0.6742	-0.0374
CNN + Color + Shape	0.7439	0.7513	-0.0066	0.6040	0.6701	0.0080
CNN + Color + Texture	0.7807	0.7919	0.0387	**0.7318**	**0.7597**	-0.0223
CNN + Color + Spatial	**0.7894**	**0.8020**	0.0358	0.7085	0.7513	-0.0170
CNN + Color + Text	0.7880	0.8023	0.0413	0.6997	0.7457	-0.0226
CNN + Shape + Texture	0.7065	0.7338	-0.0021	0.5333	0.6090	-0.0065
CNN + Shape + Spatial	0.7066	0.7240	-0.0032	0.5241	0.6036	-0.0261
CNN + Texture + Spatial	0.7302	0.7561	0.0136	0.6269	0.6856	-0.0020
CNN + Color + Shape + Texture	0.7484	0.7526	-0.0091	0.5939	0.6688	0.0084
CNN + Color + Shape + Spatial	0.7460	0.7606	-0.0066	0.6127	0.6737	0.0081
Color + Shape + Texture + Spatial	0.6956	0.7176	-0.0027	0.5434	0.6290	-0.0011
CNN + Color + Shape + Texture + Spatial	0.7495	0.7557	-0.0091	0.5964	0.6717	0.0086
CNN + Color + Shape + Texture + Spatial + Text	0.7546	0.7615	-0.0094	0.6151	0.6849	0.0032

grouped product recognition within supermarket settings. Although spatial information alone generated limited clustering performance, combining it with other features improved the results. For example, on the Grocery Products dataset, combining color with CNN resulted in an NMI of 0.7989, and combining it with CNN and Color (CNN + Color + Spatial) further increased the NMI to 0.8020. It suggests that spatial relationships are more beneficial for clustering methods when combined with visual features.

Figure 11 illustrates the clustering performance across different distance thresholds for Agglomerative Clustering and xi thresholds for OPTICS in different feature combinations. In Agglomerative Clustering, CNN-based combinations, particularly CNN + Color + Texture and CNN + Color + Spatial, achieved peak ARI and NMI scores across a range of distance thresholds.

Under OPTICS, a general decline in ARI and NMI was observed with increasing xi thresholds, showing the high sensitivity of density-based clustering methods to parameter changes. Silhouette Score remained relatively low across thresholds for both clustering methods.

Figure 12 shows examples of grouped product recognition results using Agglomerative Clustering. On the Grocery Products dataset (a)-(h), products are densely organized on shelves, with highly consistent visual features such as color and packaging design, showing that Agglomerative Clustering performs effectively under conditions where shelf organization and product appearance are highly consistent. While clustering performs reasonably well under these conditions, some incorrect predictions are still observed, as indicated by red solid bounding boxes.

In contrast, the WebMarket dataset examples (i)–(p)

Table 9: Performance on the WebMarket dataset [14] using different feature combinations

Feature	Agglomerative Clustering			OPTICS		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.6582	0.8014	-0.0188	0.4498	0.7479	-0.0309
Color	0.4561	0.6975	0.0191	0.3286	0.7220	-0.1360
Shape	0.4874	0.7009	-0.0257	0.2370	0.6536	-0.0219
Texture	0.0552	0.0229	-0.0012	0.0022	0.1835	0.0095
Spatial	0.0451	0.2351	0.0298	0.5373	0.7496	-0.2638
Text	0.0074	0.1343	0.9473	0.0064	0.1227	0.7607
CNN + Color	0.6692	0.8095	-0.0306	0.4671	0.7669	-0.0683
CNN + Shape	0.5546	0.7387	-0.0334	0.3420	0.7033	-0.0022
CNN + Texture	0.6457	0.8016	-0.0203	0.4105	0.7387	-0.0066
CNN + Spatial	0.6566	0.8011	-0.0204	0.4586	0.7512	-0.0307
CNN + Text	0.6384	0.7932	-0.0199	0.4152	0.7405	-0.0296
Color + Shape	0.6029	0.7633	-0.0345	0.3119	0.7033	-0.0301
Color + Texture	0.4583	0.6917	0.0106	0.2384	0.6918	-0.0895
Color + Spatial	0.4587	0.7009	0.0137	0.3586	0.7338	-0.1342
CNN + Color + Shape	0.6299	0.7816	-0.0525	0.4226	0.7442	-0.0212
CNN + Color + Texture	**0.6858**	**0.8183**	-0.0320	0.4520	0.7652	-0.0521
CNN + Color + Spatial	0.6754	0.8119	-0.0301	**0.4713**	**0.7683**	-0.0678
CNN + Color + Text	0.6755	0.8132	-0.0235	0.4486	0.7705	-0.0681
CNN + Shape + Texture	0.5551	0.7356	-0.0313	0.3206	0.6933	-0.0052
CNN + Shape + Spatial	0.5587	0.7416	-0.0313	0.3448	0.7026	-0.0023
CNN + Texture + Spatial	0.6506	0.8041	-0.0199	0.4043	0.7399	0.0137
CNN + Color + Shape + Texture	0.6356	0.7799	-0.0510	0.4297	0.7489	-0.0192
CNN + Color + Shape + Spatial	0.6351	0.7832	-0.0488	0.4226	0.7442	-0.0212
Color + Shape + Texture + Spatial	0.6017	0.7588	-0.0371	0.3291	0.7102	-0.0316
CNN + Color + Shape + Texture + Spatial	0.6368	0.7811	-0.0477	0.4311	0.7481	-0.0191
CNN + Color + Shape + Texture + Spatial + Text	0.6393	0.7844	-0.0506	0.4276	0.7510	0.0196

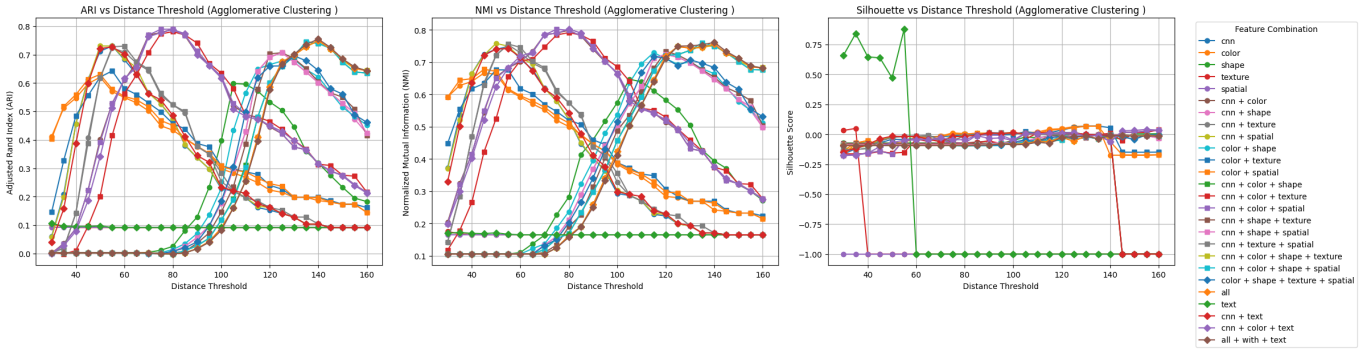
show more complex scenarios, including greater variability in product design and shelf organization. Although the model achieves reasonable grouping performance, several misgroupings were present, marked by red solid bounding boxes, reflecting the challenges of unstructured retail environments.

4.6 Optimal Threshold Selection Results

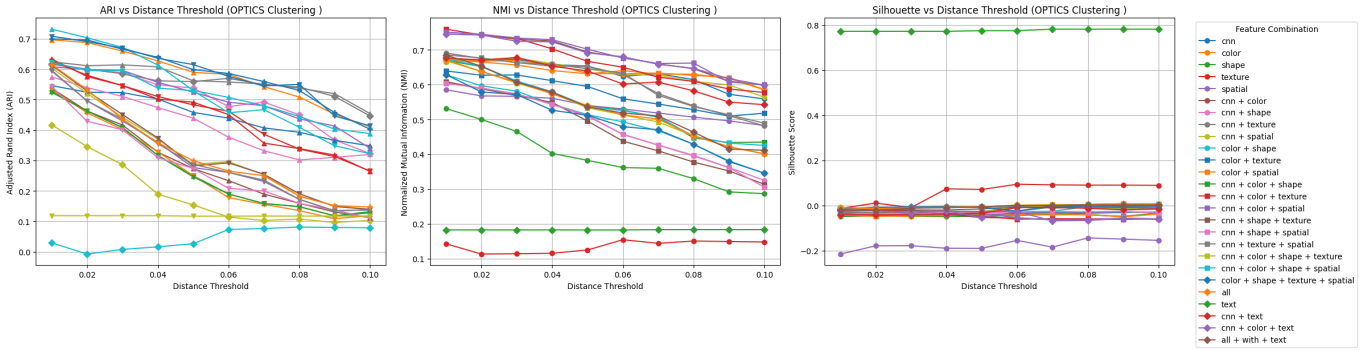
This section presents the performance results using the optimal threshold for each image selected by Silhouette Score and entropy under Agglomerative Clustering and OPTICS. Table 10 and 11 summarize the results under Agglomerative Clustering, while Table 12 and 13 report the clustering results under OPTICS.

Under Agglomerative Clustering, as shown in Table 10, on the Grocery Products dataset, using CNN alone under Silhouette Score selection gave better results than other single features, with an ARI of 0.5657 and NMI of 0.6205. When combining features, the combination of CNN + Text achieved the highest ARI (0.5897) and NMI (0.6336), outperforming other feature combinations. Other good performances, such as CNN + Color + Texture and CNN + Color + Spatial, also achieved competitive results. Under entropy selection, among single features, Color alone achieved the highest results, reaching an ARI of 0.3189 and NMI of 0.3407. This differs from Silhouette Score selection. The best performance was obtained by the combination of CNN + Color + Shape + Texture + Spatial + Text, with an ARI of 0.3797 and NMI of 0.4283.

Table 11 reports the results for the WebMarket dataset. Under Silhouette Score selection, CNN features alone reported the best ARI and NMI scores compared to other single features. Moreover, among all combina-



(a) Clustering performance under different distance thresholds using Agglomerative Clustering



(b) Clustering performance under different xi thresholds using OPTICS

Figure 11: Comparison of clustering performance across feature combinations and clustering thresholds using Agglomerative Clustering and OPTICS on the Grocery Products dataset [13]. Metrics include Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Silhouette Score.

tion features, the best overall performance was achieved by the combination of CNN + Color + Shape + Texture + Spatial, which reported an ARI of 0.6061 and an NMI of 0.7596. Under entropy selection, CNN + Color + Shape reached the highest ARI (0.2683), while CNN + Color + Shape + Texture + Spatial + Text reported the best NMI (0.5458).

For OPTICS, as we can observe in Table 12, on the Grocery Products dataset, using color features alone under Silhouette Score selection achieved the best performance with an ARI of 0.5109 and NMI of 0.6203 among individual features, followed by CNN features. The best overall performance came from CNN + Color + Spatial, with the highest ARI (0.6545) and NMI (0.7122). This outperformed all other feature combinations. Under entropy-based threshold selection, text features alone reported the best ARI (0.2566), and the combination of CNN, color, and texture gained the highest NMI (0.4778).

Table 13 presents the results for the WebMarket dataset. Under Silhouette Score selection, CNN + Color + Spatial reached the best ARI (0.4209), and the highest NMI is reported from CNN + Color + Text. with an NMI of 0.7635. In contrast, for entropy-based optimization, spatial features achieved the best performance, with an ARI of 0.0776 and NMI of 0.5399.

The results show that selecting the optimal threshold based on the Silhouette Score led to better overall performance compared to using entropy.

Figure 13 illustrates grouped product recognition results on both the Grocery Products and WebMarket datasets using dynamically selected thresholds per image. In the Grocery Products dataset, many product instances are correctly grouped, while some errors are observed. In contrast, results in the WebMarket dataset show a greater number of misgroupings.

4.7 Grouped Product Recognition Results on Dutch Markets dataset

To further evaluate the method under real-world conditions, Table 14 presents the grouped product recognition results using clustering methods under two different YOLOv5-based detection models. We selected the two top feature combinations: CNN + Color + Texture and CNN + Color + Spatial. These combinations were used based on the best results obtained in the previous recog-

Figure 12: Examples of grouped product recognition results on different datasets. (a)–(h) show examples from the Grocery Products dataset [13], and (i)–(p) illustrate examples from the WebMarket dataset [14]. Red solid bounding boxes are used to indicate prediction errors.

nition experiments on Grocery Products and WebMarket datasets (see Table 8 and Table 9).

The results show that the best overall recognition performance was achieved by applying Agglomerative Clustering with the detection model trained on the Grocery Products dataset. When using CNN + Color + Texture,

Table 10: Performance of grouped product recognition on the Grocery Products dataset [13] using per-image optimal thresholds selected by Silhouette Score and entropy under Agglomerative Clustering.

Feature	Silhouette Score			Entropy		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.5657	0.6205	-0.0716	0.2693	0.3133	0.0011
Color	0.5178	0.5342	-0.2222	0.3189	0.3407	0.0155
Shape	0.3114	0.3664	-0.0279	0.2039	0.2736	-0.0164
Texture	0.1424	0.2271	-0.0291	0.1515	0.2279	-0.0085
Spatial	0.0930	0.1664	0.4903	0.0931	0.1664	0.4903
Text	0.1137	0.1741	0.7725	0.1092	0.1694	0.7104
CNN + Color	0.5829	0.5801	-0.1228	0.3665	0.3662	0.0162
CNN + Shape	0.4898	0.5680	-0.0382	0.2656	0.3192	-0.0034
CNN + Texture	0.5408	0.5976	-0.0602	0.2789	0.3228	-0.0193
CNN + Spatial	0.5689	0.6237	-0.0801	0.2757	0.3197	0.0017
CNN + Text	**0.5897**	**0.6336**	-0.0758	0.2821	0.3204	0.0059
Color + Shape	0.4379	0.4542	-0.0344	0.3584	0.3899	0.0028
Color + Texture	0.4119	0.4250	-0.1280	0.3304	0.3473	0.0074
Color + Spatial	0.5069	0.5274	-0.2234	0.3380	0.3498	0.0065
CNN + Color + Shape	0.5466	0.5948	-0.0311	0.3735	0.4240	-0.0036
CNN + Color + Texture	0.5859	0.6008	-0.0951	0.3684	0.3660	0.0127
CNN + Color + Spatial	0.5796	0.5770	-0.1226	0.3635	0.3631	0.0133
CNN + Color + Text	0.5685	0.5843	-0.1234	0.3603	0.3622	0.0204
CNN + Shape + Texture	0.4690	0.5491	-0.0298	0.2554	0.3139	0.0014
CNN + Shape + Spatial	0.4898	0.5680	-0.0385	0.2656	0.3192	-0.0034
CNN + Texture + Spatial	0.5408	0.5976	-0.0569	0.2789	0.3228	-0.0193
CNN + Color + Shape + Texture	0.5466	0.5948	-0.0298	0.3735	0.4240	-0.0011
CNN + Color + Shape + Spatial	0.5466	0.5948	-0.0311	0.3735	0.4240	-0.0036
Color + Shape + Texture + Spatial	0.4183	0.4397	-0.0356	0.3439	0.3829	-0.0021
CNN + Color + Shape + Texture + Spatial	0.5466	0.5948	-0.0299	0.3797	0.4283	0.0022
CNN + Color + Shape + Texture + Spatial + Text	0.5596	0.6002	-0.0310	**0.3797**	**0.4283**	0.0024

it reached the highest ARI of 0.6140 and NMI of 0.6638. The second-best result was CNN + Color + Spatial, with an ARI of 0.6082. In contrast, OPTICS obtained lower performance across both feature combinations.

In comparison, when using the YOLOv5 model trained on the WebMarket dataset, recognition performance was slightly lower under the same clustering methods. Under Agglomerative Clustering, the best combination, CNN + Color + Spatial, reported an ARI of 0.5887 and an NMI of 0.6438. Moreover, compared to Agglomerative Clustering, the performance under OPTICS was lower across all feature combinations.

Figure 14 illustrates grouped product recognition example results on the Dutch Markets dataset. In structured and less cluttered settings such as (a) and (c), the model achieved accurate and consistent grouping. Nevertheless, challenges remain in more complex scenarios. In examples like (d), (g), and (h), products are densely packed and share high visual similarity, leading to misgroupings presented with red bounding boxes.

We also evaluated how optimal threshold selection affects grouping performance. For this experiment, we used the YOLOv5 model trained on the Grocery Products dataset as the detector since it showed higher and more stable performance across previous evaluations (see Table 14). As shown in Table 15 and 16, thresholds were chosen based on Silhouette Score and entropy. The results show that OPTICS outperformed Agglomerative Clustering, achieving the highest ARI of 0.4968 and NMI of 0.5891 using the CNN + Color + Spatial combination under Silhouette Score selection. In contrast, the best result under Agglomerative Clustering was obtained using the

Table 11: Performance of grouped product recognition on the WebMarket dataset [14] using per-image optimal thresholds selected by Silhouette Score and entropy under Agglomerative Clustering.

Feature	Silhouette Score			Entropy		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.5368	0.7395	-0.0818	0.0535	0.2656	0.0076
Color	0.0712	0.3719	-0.2561	0.0368	0.2377	-0.0254
Shape	0.3989	0.6425	-0.0391	0.1052	0.3414	-0.0380
Texture	0.0533	0.2651	-0.0742	0.0359	0.1896	-0.0273
Spatial	0.0823	0.3091	-0.0907	0.0533	0.2418	-0.0771
Text	0.0074	0.1345	0.6787	0.0056	0.1214	0.6948
CNN + Color	0.3871	0.6802	-0.0960	0.0944	0.3470	0.0014
CNN + Shape	0.4532	0.6790	-0.0554	0.1749	0.4416	-0.0213
CNN + Texture	0.5411	0.7364	-0.0751	0.0572	0.2604	0.0057
CNN + Spatial	0.5501	0.7468	-0.0823	0.0583	0.2633	0.0061
CNN + Text	0.5605	0.7529	-0.0875	0.0533	0.2633	0.0082
Color + Shape	0.5084	0.7063	-0.0687	0.2026	0.4703	-0.0255
Color + Texture	0.0651	0.3591	-0.1983	0.0407	0.2472	-0.0240
Color + Spatial	0.0672	0.3674	-0.2442	0.0355	0.2354	-0.0253
CNN + Color + Shape	0.6057	0.7540	-0.0699	**0.2683**	0.5437	-0.0024
CNN + Color + Texture	0.3751	0.6726	-0.0977	0.0951	0.3473	0.0008
CNN + Color + Spatial	0.4545	0.7027	-0.0981	0.1048	0.3628	0.0242
CNN + Color + Text	0.3938	0.6872	-0.0979	0.0954	0.3492	0.0002
CNN + Shape + Texture	0.4803	0.6858	-0.0437	0.1745	0.4447	-0.0264
CNN + Shape + Spatial	0.4608	0.6824	-0.0534	0.1745	0.4413	-0.0214
CNN + Texture + Spatial	0.5378	0.7384	-0.0734	0.0579	0.2617	0.0068
CNN + Color + Shape + Texture	0.5958	0.7563	-0.0595	0.2596	0.5437	-0.0270
CNN + Color + Shape + Spatial	0.6057	0.7540	-0.0731	0.2587	0.5411	-0.0228
Color + Shape + Texture + Spatial	0.5105	0.7076	-0.0601	0.2271	0.4857	-0.0224
CNN + Color + Shape + Texture + Spatial	**0.6061**	**0.7596**	-0.0608	0.2599	0.5425	-0.0268
CNN + Color + Shape + Texture + Spatial + Text	0.5590	0.7571	-0.0512	0.2618	**0.5458**	-0.0337

CNN + Color + Shape + Texture + Spatial + Text combination, with an ARI of 0.3534 and NMI of 0.3883, also under Silhouette Score-based thresholds.

When using entropy-based threshold selection, the overall performance was lower for both clustering methods. Under Agglomerative Clustering, CNN + Color + Shape + Texture + Spatial + Text feature combination reported the best result, with an ARI of 0.1853 and NMI of 0.2389. In comparison, the Color feature alone obtained the best performance, reaching an ARI of 0.2194 and NMI of 0.4789.

4.8 Impact of Product Size on Clustering Performance

Since product size may affect the visibility of key features such as text or shape, we analyzed whether different sizes impact clustering performance. All products were divided into three size groups based on bounding box area: small, medium, and large.

As illustrated in Figure 15, it shows clearly different distributions of product sizes across the three datasets, computed as the bounding box area divided by the entire image area. Two vertical dashed lines represent fixed thresholds used to categorize sizes. the red line at 0.025 separates small and medium products, while the green line at 0.08 separates medium and large products. The Grocery Products dataset presents a relatively balanced distribution, especially with more medium and large items. This shows that the Grocery Products dataset contains varied product scales. In contrast, most products in the WebMarket dataset are small. The majority of bounding boxes are smaller than the 0.025 threshold,

Table 12: Performance of grouped product recognition on the Grocery Products dataset [13] using per-image optimal thresholds selected by Silhouette Score and entropy under OPTICS.

Feature	Silhouette Score			Entropy		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.4798	0.6036	-0.0300	0.1353	0.4024	-0.0282
Color	0.5109	0.6203	-0.0735	0.2434	0.4695	-0.0855
Shape	0.2057	0.4243	-0.0524	0.1137	0.3096	-0.0014
Texture	-0.0007	0.1028	-0.0251	0.0703	0.1287	-0.0070
Spatial	0.4950	0.5557	-0.2173	**0.2566**	0.4347	-0.1993
Text	0.1151	0.1807	0.7462	0.1168	0.1810	0.7457
CNN + Color	0.6699	0.7071	-0.0491	0.2067	0.4773	-0.0330
CNN + Shape	0.3259	0.5127	-0.0438	0.1031	0.3423	-0.0133
CNN + Texture	0.3443	0.4882	-0.0121	0.0997	0.3354	-0.0130
CNN + Spatial	0.4902	0.6203	-0.0342	0.1435	0.4037	-0.0443
CNN + Text	0.4925	0.6072	-0.0343	0.1417	0.3833	-0.0316
Color + Shape	0.3003	0.5209	-0.0165	0.1158	0.3942	-0.0136
Color + Texture	0.4418	0.5639	-0.0660	0.2255	0.3927	-0.0430
Color + Spatial	0.5655	0.6505	-0.0640	0.2250	0.4611	-0.0789
CNN + Color + Shape	0.3524	0.5672	-0.0154	0.1304	0.3942	-0.0136
CNN + Color + Texture	0.5750	0.6372	-0.0285	0.2062	0.4712	-0.0283
CNN + Color + Spatial	**0.6545**	**0.7122**	-0.0683	0.2057	0.4717	-0.0326
CNN + Color + Text	0.6432	0.6981	-0.0505	0.2155	**0.4778**	-0.0309
CNN + Shape + Texture	0.2861	0.4921	-0.0280	0.1065	0.3445	-0.0057
CNN + Shape + Spatial	0.3109	0.5067	-0.0435	0.0945	0.3356	-0.0140
CNN + Texture + Spatial	0.3738	0.5008	-0.0093	0.0977	0.3400	-0.0122
CNN + Color + Shape + Texture	0.3608	0.5735	-0.0135	0.1276	0.3951	-0.0054
CNN + Color + Shape + Spatial	0.3553	0.5640	-0.0153	0.1281	0.3924	-0.0137
Color + Shape + Texture + Spatial	0.2630	0.4975	-0.0147	0.1443	0.4177	-0.0123
CNN + Color + Shape + Texture + Spatial	0.3607	0.5735	-0.0134	0.1276	0.3952	-0.0053
CNN + Color + Shape + Texture + Spatial + Text	0.3722	0.5784	-0.0146	0.1371	0.4069	-0.0046

with few medium and almost no large items. The Dutch Markets dataset shows a broader distribution, with many products falling within the medium range. It offers a relatively balanced dataset of small, medium, and large products.

Table 17 compares the clustering performance of Agglomerative Clustering across three datasets based on product size (small, medium, large). In the Grocery Product dataset, medium-sized products achieved the highest ARI (0.8494), showing that these products were grouped more accurately. In contrast, the WebMarket dataset reached perfect ARI and NMI for large products, which is likely due to the small number of large-sized samples. For the Dutch Markets dataset, medium products show the best performance, with an ARI of 0.6451 and NMI of 0.6671, followed by large-sized items. Small products show weaker results. The results suggest that the Grocery Products and Dutch Markets dataset, with a more balanced size distribution, supports better clustering overall, while most items in the WebMarket dataset are small, clustering becomes more difficult. This may be due to the fact that small products often lack clear visual information, such as readable text or distinct shapes, leading to lower clustering accuracy.

5 Discussion

This section presents a detailed discussion of the experimental results based on the research questions, and also discusses the proposed framework limitations and future improvements.

Table 13: Performance of grouped product recognition on the WebMarket dataset [14] using per-image optimal thresholds selected by Silhouette Score and entropy under OPTICS.

Feature	Silhouette Score			Entropy		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.3647	0.7313	-0.0744	0.0044	0.2823	-0.0527
Color	0.2692	0.7079	-0.1785	0.0530	0.5341	-0.2101
Shape	0.0848	0.5517	-0.0597	-0.0001	0.1645	-0.0127
Texture	-0.0028	0.1018	-0.0501	-0.0036	0.1080	-0.0199
Spatial	0.5195	0.7408	-0.2897	**0.0776**	**0.5399**	-0.3912
Text	0.0064	0.1227	0.7307	0.0060	0.1204	0.6729
CNN + Color	0.4171	0.7606	-0.1067	0.0177	0.4051	-0.1065
CNN + Shape	0.2928	0.6841	-0.0448	0.0003	0.1853	-0.0185
CNN + Texture	0.3375	0.7243	-0.0676	-0.0003	0.1447	0.0108
CNN + Spatial	0.3717	0.7333	-0.0737	0.0047	0.2853	0.0528
CNN + Text	0.3402	0.7261	-0.0800	0.0043	0.2682	-0.0547
Color + Shape	0.2309	0.6834	-0.0643	0.0000	0.2020	-0.0091
Color + Texture	0.2306	0.6472	-0.1562	0.0054	0.2625	-0.0608
Color + Spatial	0.3006	0.7252	-0.1808	0.0518	0.5401	-0.2073
CNN + Color + Shape	0.2851	0.7232	-0.0493	-0.0007	0.1948	-0.0061
CNN + Color + Texture	0.4052	0.7550	-0.0949	0.0094	0.3263	-0.0879
CNN + Color + Spatial	**0.4209**	0.7620	-0.1078	0.0177	0.4051	-0.1064
CNN + Color + Text	0.4015	**0.7635**	-0.1104	0.0183	0.4109	-0.1080
CNN + Shape + Texture	0.2857	0.6783	-0.0410	0.0012	0.1892	-0.0010
CNN + Shape + Spatial	0.2928	0.6841	-0.0448	0.0003	0.1853	-0.0184
CNN + Texture + Spatial	0.3342	0.7243	-0.0697	-0.0003	0.1447	0.0105
CNN + Color + Shape + Texture	0.2801	0.7246	-0.0518	-0.0003	0.2058	-0.0091
CNN + Color + Shape + Spatial	0.2851	0.7232	-0.0493	-0.0007	0.1948	-0.0057
Color + Shape + Texture + Spatial	0.2387	0.6868	-0.0616	-0.0007	0.1790	-0.0103
CNN + Color + Shape + Texture + Spatial	0.2801	0.7246	-0.0518	-0.0003	0.2058	-0.0088
CNN + Color + Shape + Texture + Spatial + Text	0.2870	0.7267	-0.0530	-0.0001	0.2068	-0.0089

5.1 Answers to Research Questions

What visual features, such as color and texture, are most effective for grouping retail products?

The results clearly show that CNN-based features capture rich semantic representations, outperforming traditional descriptors such as color and texture when used alone. As shown in Table 8 and 9, the combination of CNN-based features with color features brought significant improvements, showing that combining different types of visual features helps the model group products more accurately.

How does textual information extracted from product packaging (e.g., labels, brand names) affect grouped product recognition?

Textual features played a supplementary role in grouped product recognition, with distinct behaviors under fixed and optimal threshold conditions. Under a fixed threshold, the results demonstrate little improvement or even negative effects on ARI and NMI, though they improved the Silhouette Score in some feature combinations. This can be attributed to challenges such as inconsistent text orientation and frequent occlusions in packaging. Image quality is also a key issue. Blurry images make it difficult for the model to detect and recognize text accurately.

For instance, in the Grocery Product dataset, adding text features to CNN and color dropped ARI from 0.7894 to 0.7880, and only a small NMI gain was observed. While in the WebMarket dataset, ARI slightly increased from 0.6692 to 0.6755, and the NMI from 0.8095 to 0.8132. This suggests that textual features had inconsistent effects and may even introduce noise when combined with strong visual features.

Figure 13: Examples of grouped product recognition results using optimized thresholds per image obtained from Silhouette Score on two datasets. (a)–(h) show examples from Grocery Products dataset [13], and (i)–(p) illustrate examples from WebMarket dataset [14]. Red solid bounding boxes are used to indicate prediction errors.

While under the optimal threshold setting, the effect of textual features became more positive. In the Grocery Products dataset, adding text to CNN gave the best ARI (0.5897) and NMI (0.6336) among all combinations, suggesting that text contributed meaningfully to clustering when combined with CNN. In the WebMarket

Feature	Agglomerative Clustering			OPTICS		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
YOLOv5 - Grocery Products						
CNN + Color + Texture	0.6140	0.6638	0.0821	0.5413	0.6034	-0.0277
CNN + Color + Spatial	0.6082	0.6611	0.0861	0.5423	0.6042	-0.0372
YOLOv5 - WebMarket						
CNN + Color + Texture	0.5780	0.6421	0.0620	0.5098	0.5848	-0.0442
CNN + Color + Spatial	0.5887	0.6438	0.0682	0.5242	0.5920	-0.0456

Table 14: Performance of grouped product recognition on Dutch Markets dataset using top-2 combined features with two YOLOv5-based detection models.

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 14: Predictions of grouped product recognition on Dutch Markets dataset. Red solid bounding boxes denote incorrect predictions.

dataset, adding text features to CNN increased ARI from 0.5368 to 0.5605 and NMI from 0.7395 to 0.7529. However, adding text to a strong feature combination (CNN + Color + Shape + Texture + Spatial) under Silhouette Score selection led to a weaker performance, reducing the ARI from 0.6061 to 0.5590 and the NMI from 0.7596 to 0.7571, indicating that text may sometimes introduce noise.

These results show that text features can help recognition performance when the threshold is carefully selected, but they do not always work in every case. For example, when the image quality is low or the visual features are already strong, adding text may reduce performance.

How do spatial relationships between products affect unsupervised grouping in supermarket environments?

Spatial features show small but consistent improvements in product grouping performance, especially when items from the same brand or category are placed close together.

Table 15: Performance of grouped product recognition on the Dutch Markets dataset using per-image optimal thresholds selected by Silhouette Score and entropy under Agglomerative Clustering.

Feature	Silhouette Score			Entropy		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.3088	0.3563	-0.0352	0.1306	0.1792	0.0112
Color	0.2763	0.2872	-0.1346	0.1428	0.1695	-0.2344
Shape	0.2628	0.3237	-0.0382	0.1116	0.1544	-0.0094
Texture	0.1061	0.1469	-0.0400	0.0978	0.1436	-0.0432
Spatial	0.0470	0.0751	-0.0501	0.0470	0.0751	-0.0501
Text	0.0767	0.1096	0.3505	0.0677	0.0998	0.5231
CNN + Color	0.3304	0.3566	-0.0613	0.1371	0.1748	0.0333
CNN + Shape	0.2821	0.3453	-0.0207	0.1217	0.1714	-0.0095
CNN + Texture	0.2921	0.3436	-0.0463	0.1298	0.1775	-0.0170
CNN + Spatial	0.3085	0.3566	-0.0317	0.1316	0.1806	0.0108
CNN + Text	0.3099	0.3567	-0.0207	0.1309	0.1780	-0.0057
Color + Shape	0.2983	0.3338	-0.0549	0.1501	0.1896	0.0039
Color + Texture	0.2127	0.2363	-0.0627	0.1390	0.1692	-0.2089
Color + Spatial	0.2757	0.2877	-0.1464	0.1435	0.1709	0.0728
CNN + Color + Shape	0.3438	0.3839	-0.0269	0.1813	0.2343	0.0001
CNN + Color + Texture	0.3046	0.3235	-0.0733	0.1381	0.1753	0.0302
CNN + Color + Spatial	0.3339	0.3644	-0.0456	0.1380	0.1753	0.0259
CNN + Color + Text	0.3315	0.3644	-0.0564	0.1394	0.1772	0.0358
CNN + Shape + Texture	0.2864	0.3501	-0.0177	0.1233	0.1783	-0.0060
CNN + Shape + Spatial	0.2800	0.3404	-0.0207	0.1216	0.1712	-0.0095
CNN + Texture + Spatial	0.2889	0.3404	-0.0485	0.1302	0.1781	-0.0171
CNN + Color + Shape + Texture	0.3515	0.3835	-0.0279	0.1847	0.2378	0.0006
CNN + Color + Shape + Spatial	0.3436	0.3840	-0.0216	0.1808	0.2343	-0.0001
Color + Shape + Texture + Spatial	0.3006	0.3445	-0.0118	0.1523	0.1920	0.0049
CNN + Color + Shape + Texture + Spatial	0.3474	0.3810	-0.0299	0.1847	0.2378	0.0002
CNN + Color + Shape + Texture + Spatial + Text	**0.3534**	**0.3883**	-0.0276	**0.1853**	**0.2389**	-0.0039

In the WebMarket dataset, adding spatial features to CNN and color increased ARI from 0.6692 to 0.6754 and NMI from 0.8095 to 0.8119. This shows that spatial information can give extra support to the clustering process, even in less organized shelf scenarios. The Grocery Products dataset showed no change in ARI and a small increase in NMI. These findings suggest that spatial information can provide additional support to the clustering process, but its impact may be different.

How to leverage visual features, textual information, and spatial relationships to recognize and group objects in retail environments in an unsupervised manner?

To group retail products in an unsupervised way, this study combined visual, textual, and spatial features. Visual features, especially CNN-based features, gave the best performance in most cases. Color and texture added extra details and improved results when combined with CNN. Textual information extracted from packaging gave slight help. Text features were helpful in some cases when texts were visible and clear. However, when the text was blurred or occluded, they became less effective. Spatial relationships provided small improvements. By combining these features and using a clustering algorithm, the framework is able to recognize and group products based on their appearance and layout.

How effective are vision language models(VLMs) compared with the proposed framework in detecting and localizing products in dense retail environments?

In retail environments, it is important to detect and lo-

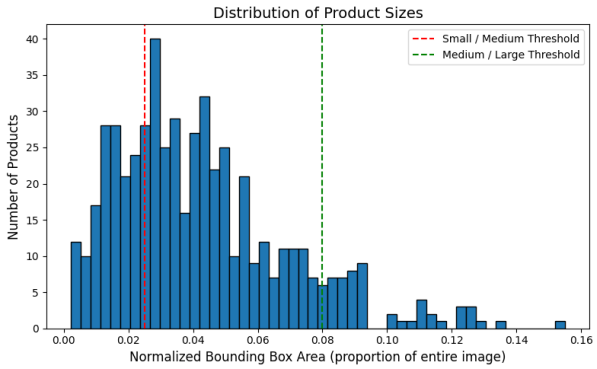
Table 16: Performance of grouped product recognition on the Dutch Markets dataset using per-image optimal thresholds selected by Silhouette Score and entropy under OPTICS.

Feature	Silhouette Score			Entropy		
	ARI	NMI	Silhouette Score	ARI	NMI	Silhouette Score
CNN	0.3858	0.5144	-0.0223	0.0977	0.3835	-0.0312
Color	0.4718	0.5698	-0.1037	**0.2194**	**0.4789**	-0.1228
Shape	0.1918	0.4018	-0.0135	0.0496	0.2829	-0.0090
Texture	0.0029	0.0828	-0.0433	0.0244	0.0678	-0.0212
Spatial	0.3718	0.4645	-0.2204	0.1884	0.3794	-0.1811
Text	0.0742	0.1071	0.7184	0.0739	0.1068	0.7149
CNN + Color	0.4925	0.5879	-0.0600	0.1693	0.4489	-0.0404
CNN + Shape	0.2470	0.4354	-0.0132	0.0499	0.2977	-0.0112
CNN + Texture	0.2839	0.4495	-0.0260	0.0676	0.3374	-0.0308
CNN + Spatial	0.4045	0.5262	-0.0218	0.0986	0.3853	-0.0315
CNN + Text	0.3633	0.5089	-0.0184	0.0975	0.3752	-0.0357
Color + Shape	0.2297	0.4263	-0.0329	0.0648	0.3255	-0.0156
Color + Texture	0.3127	0.4267	-0.0882	0.1081	0.3461	-0.0701
Color + Spatial	0.4800	0.5712	-0.1019	0.2125	0.4779	-0.1206
CNN + Color + Shape	0.3028	0.4872	-0.0143	0.0605	0.3414	-0.0163
CNN + Color + Texture	0.4545	0.5573	-0.0538	0.1240	0.4110	-0.0359
CNN + Color + Spatial	**0.4968**	**0.5891**	-0.0588	0.1691	0.4498	-0.0398
CNN + Color + Text	0.4822	0.5786	-0.0396	0.1595	0.4445	-0.0406
CNN + Shape + Texture	0.2683	0.4551	-0.0165	0.0537	0.3056	-0.0075
CNN + Shape + Spatial	0.2495	0.4355	-0.0130	0.0509	0.2979	-0.0130
CNN + Texture + Spatial	0.2869	0.4537	-0.0263	0.0695	0.3442	-0.0320
CNN + Color + Shape + Texture	0.3087	0.4962	-0.0150	0.0617	0.3460	-0.0166
CNN + Color + Shape + Spatial	0.3035	0.4874	-0.0144	0.0605	0.3414	-0.0164
Color + Shape + Texture + Spatial	0.2327	0.4317	-0.0122	0.0596	0.3158	-0.0150
CNN + Color + Shape + Texture + Spatial	0.3132	0.4977	-0.0140	0.0629	0.3371	-0.0165
CNN + Color + Shape + Texture + Spatial + Text	0.2888	0.4819	-0.0140	0.0624	0.3362	-0.0150

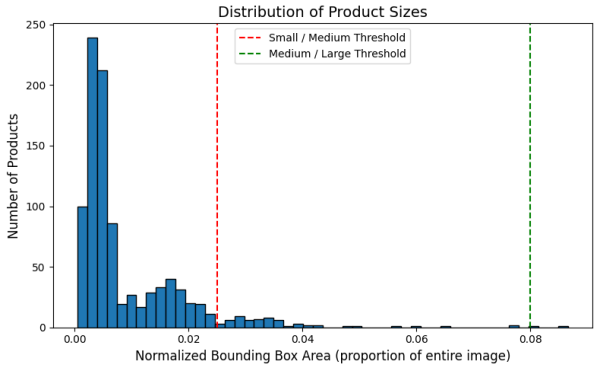
calize products accurately and clearly. However, vision language models (VLMs), such as InternVL, often struggle with detecting products accurately. Their performance varies significantly depending on the prompt used. Finding suitable prompts is challenging and time-consuming. The results show that the overall performance was low, with the best accuracy reaching only 15% (see Table 7). In contrast, our implementation, which uses YOLOv5, detected more products and showed better accuracy, as shown in Table 5. These results show that although VLMs can understand both language and images, they are still limited in accurate product detection in dense retail scenes. Furthermore, while VLMs can describe what they see in an image, they fail to localize objects correctly. This is crucial in practical scenarios, especially in robotics, where precise coordinates are essential for interacting with physical items. For example, a robot recognizes what an object is, but without accurate coordinates, it cannot grasp or move the item. Therefore, current VLMs are not yet suitable for applications that demand precise object locations.

5.2 Limitations

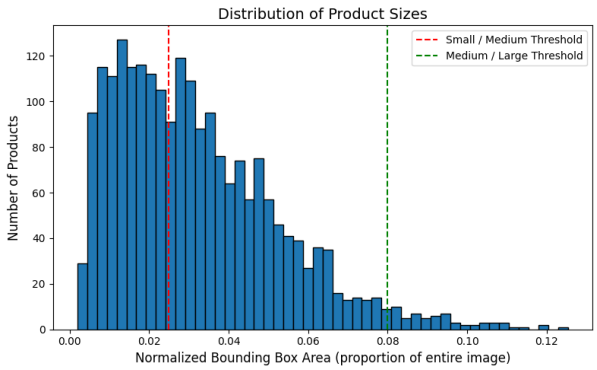
This study has several limitations. One issue is that the performance of product detection drops in cluttered or dense shelf scenarios. When shelves are heavily cluttered, products are often placed closely together or overlapping, making it difficult for the detection model to accurately separate individual items, as illustrated in Figure 16a. Second, the quality of extracted text features is impacted by image conditions. Blurry images or small fonts on packaging often lead to recognition errors. Fi-



(a) Grocery Products



(b) WebMarket



(c) Dutch Markets

Figure 15: Distribution of normalized product sizes (bounding box areas) in the Grocery Products [13], WebMarket [14], and Dutch Markets datasets. The x-axis represents the proportion of the entire image area occupied by each product. The y-axis indicates the number of products. Vertical dashed lines mark thresholds for small, medium, and large size categories.

Product size	ARI	NMI
Grocery Products [13]		
Small	0.7193	0.8437
Medium	0.8494	0.8294
Large	0.6125	0.7599
WebMarket [14]		
Small	0.7014	0.8303
Medium	0.6156	0.8060
Large	1.0000	1.0000
Dutch Markets		
Small	0.5442	0.6497
Medium	0.6451	0.6671
Large	0.6015	0.6585

Table 17: Comparison of clustering performance by product size (small, medium, large) on Agglomerative Clustering across three datasets: Grocery Products [13], WebMarket [14], and Dutch Markets.

nally, when products share very similar colors, shapes, or packaging designs, the extracted features may not be distinct enough, leading to incorrect grouping. Moreover, the clustering algorithms are sensitive to parameter settings. As shown in Figure 16b, it illustrates how visually similar juice packages were incorrectly grouped together (red bounding boxes), despite belonging to different product categories.

(a) Detection limitation (b) Clustering limitation

Figure 16: Illustration of limitations observed in detection and clustering stages.

5.3 Future Works

In the future, this framework could be improved in several ways. First, the detection model can be improved to

find small and overlapping products by using more advanced detectors such as YOLOv7 [59], or by integrating post-processing steps, for instance, planogram-based refinement [20]. Text recognition can utilize stronger OCR models or better image pre-processing to address the difficulties of blurry or low-resolution packaging. For example, TrOCR [60] applies transformer-based language modeling for more accurate text recognition. Furthermore, clustering methods could be improved by exploring techniques such as Invariant Information Clustering (IIC) [61], which may help produce more stable and meaningful groups by maximizing the consistency between different views of the same product.

6 Conclusion

This study shows that unsupervised clustering based on integrating visual, textual, and spatial features is a promising direction for grouped product recognition. The proposed method is able to cluster similar products without the need for manual labels. Experimental results demonstrate that the most effective clustering results were obtained when CNN-based visual features were used together with color features, while the contribution of text and spatial features is smaller but useful. Although the framework performs well on structured datasets like Grocery Products and shows promise on more complex datasets like WebMarket, it faces challenges when dealing with blurred images, overlapping objects, or inconsistent shelf arrangements.

Despite these limitations, the research offers a practical foundation for real-world applications such as automatic shelf monitoring, inventory management, and planogram compliance in retail settings. From an application perspective, the proposed framework demonstrates strong potential for real-world retail applications. Although some challenges remain, such as handling cluttered shelves and visually similar products, its unsupervised design eliminates the need for manual labeling, significantly reducing annotation costs and making it well-suited for large-scale implementation. With further improvements, this framework offers a feasible path toward real-world implementation.

Appendix A Prompts

- **P1:**
  *"Please detect each *individual* retail product in this image and return the bounding box coordinates for every detected product as a list of [x_min, y_min, x_max, y_max]. Each bounding box should correspond to a *separate* product instance."*

- **P2:**
  *"Detect and return all *individual product* bounding boxes in this image. Format: a list of bounding boxes where each box is represented as [x_min, y_min, x_max, y_max]. Example output: [[100, 200, 300, 400], [320, 210, 510, 400], ...] Each bounding box should represent one separate product on the shelf."*

- **P3:**
  "Please detect each individual retail product in this image. For each product, return only the bounding box as a 4-number list in the format [x_min, y_min, x_max, y_max]. Do NOT include any description, explanation, or try to draw the boxes—just list the coordinates. Example output:[100, 150, 200, 250], [220, 160, 320, 260]..."

- **P4:**
  "Let's assume the image is 640x640, follow an object detection approach to identify all the instances in the image, and generate their object bounding boxes. Give me the bounding boxes in a relative format (0 to 1), I don't need the code, just the bboxes. The result must be in this format: [[x1, x2, y1, y2] ...]"

- **P5:**
  "Assume the image is 640x640. Please return all visible retail product bounding boxes in normalized coordinates (0–1), following the format [x1, x2, y1, y2]."

- **P6:**
  "Assume the image is 640x640. Please return all visible retail product bounding boxes in normalized coordinates (0–1), following the format [x1, x2, y1, y2]."

References

[1] Y. Wei, S. Tran, S. Xu, B. Kang, and M. Springer, "Deep learning for retail product recognition: Challenges and techniques," *Computational intelligence and neuroscience*, vol. 2020, no. 1, p. 8875910, 2020.

[2] C. G. Melek, E. B. Sönmez, and S. Varlı, "Datasets and methods of product recognition on grocery shelf images using computer vision and machine learning approaches: An exhaustive literature review,"

Engineering Applications of Artificial Intelligence, vol. 133, p. 108452, 2024.

[3] A. Tonioni and L. di Stefano, "Product recognition in store shelves as a sub-graph isomorphism problem. corr abs/1707.08378 (2017)," *arXiv preprint arXiv:1707.08378*, 2017.

[4] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "Scrdet: Towards more robust detection for small, cluttered and rotated objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[5] I. Baz, E. Yoruk, and M. Cetin, "Context-aware hybrid classification system for fine-grained retail product recognition," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pp. 1–5, IEEE, 2016.

[6] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[7] L. Xiao, X. Yang, X. Lan, Y. Wang, and C. Xu, "Towards visual grounding: A survey," *arXiv preprint arXiv:2412.20206*, 2024.

[8] K. Fuchs, T. Grundmann, and E. Fleisch, "Towards identification of packaged products via computer vision: Convolutional neural networks for object detection and image classification in retail environments," in *Proceedings of the 9th International Conference on the Internet of Things*, pp. 1–8, 2019.

[9] X. Jiang, A. Hadid, Y. Pang, E. Granger, and X. Feng, "Deep learning in object detection and recognition," 2019.

[10] P. Selvam and J. A. S. Koilraj, "A deep learning framework for grocery product detection and recognition," *Food Analytical Methods*, vol. 15, no. 12, pp. 3498–3522, 2022.

[11] M. George, D. Mircic, G. Soros, C. Floerkemeier, and F. Mattern, "Fine-grained product class recognition for assisted shopping," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015.

[12] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[13] M. George and C. Floerkemeier, "Recognizing products: A per-exemplar multi-label image classification approach," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*, pp. 440–455, Springer, 2014.

[14] Y. Zhang, L. Wang, R. Hartley, and H. Li, "Where's the weet-bix?," in *Computer Vision–ACCV 2007: 8th Asian Conference on Computer Vision, Tokyo, Japan, November 18-22, 2007, Proceedings, Part I 8*, pp. 800–810, Springer, 2007.

[15] A. Tonioni, E. Serra, and L. Di Stefano, "A deep learning pipeline for product recognition on store shelves," in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, pp. 25–31, 2018.

[16] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[18] E. Gothai, S. Bhatia, A. Alabdali, D. Sharma, B. Raj, and P. Dadheech, "Design features of grocery product recognition using deep learning," *Intelligent Automation and Soft Computing*, vol. 34, no. 2, pp. 1231–1246, 2022.

[19] R. Litman, O. Anschel, S. Tsiper, R. Litman, S. Mazor, and R. Manmatha, "Scatter: Selective context attentional scene text recognizer," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11959–11969, 2020.

[20] P. Selvam, M. Faheem, V. Dakshinamurthi, A. Nevgi, R. Bhuvaneswari, K. Deepak, and J. Abraham Sundar, "Batch normalization free rigorous feature flow neural network for grocery product recognition," *IEEE Access*, vol. 12, pp. 68364–68381, 2024.

[21] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5551–5560, 2017.

[22] O. E. Olorunshola, M. E. Irhebhude, and A. E. Evwiekpaefe, "A comparative study of yolov5 and

yolov7 object detection algorithms," *Journal of Computing and Social Informatics*, vol. 2, no. 1, pp. 1–12, 2023.

[23] M. G. Ragab, S. J. Abdulkader, A. Muneer, A. Alqushaibi, E. H. Sumiea, R. Qureshi, S. M. Al-Selwi, and H. Alhussian, "A comprehensive systematic review of yolo for medical object detection (2018 to 2023)," *IEEE Access*, 2024.

[24] B. Mahaur and K. Mishra, "Small-object detection based on yolov5 in autonomous driving systems," *Pattern Recognition Letters*, vol. 168, pp. 115–122, 2023.

[25] U. Nepal and H. Eslamiat, "Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs," *Sensors*, vol. 22, no. 2, p. 464, 2022.

[26] R. Khanam and M. Hussain, "What is yolov5: A deep look into the internal features of the popular object detector. arxiv 2024," *arXiv preprint arXiv:2407.20892*.

[27] S. Li, S. Liu, Z. Cai, Y. Liu, G. Chen, and G. Tu, "Tc-yolov5: rapid detection of floating debris on raspberry pi 4b," *Journal of Real-Time Image Processing*, vol. 20, no. 2, p. 17, 2023.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[29] D. Theckedath and R. Sedamkar, "Detecting affect states using vgg16, resnet50 and se-resnet50 networks," *SN Computer Science*, vol. 1, no. 2, p. 79, 2020.

[30] M. Elpeltagy and H. Sallam, "Automatic prediction of covid- 19 from chest images using modified resnet50," *Multimedia tools and applications*, vol. 80, no. 17, pp. 26451–26463, 2021.

[31] V. Vinaykumar, J. A. Babu, and J. Frnda, "Optimal guidance whale optimization algorithm and hybrid deep learning networks for land use land cover classification," *EURASIP Journal on Advances in Signal Processing*, vol. 2023, no. 1, p. 13, 2023.

[32] S. Ray, "Disease classification within dermascopic images using features extracted by resnet50 and classification through deep forest," *arXiv preprint arXiv:1807.05711*, 2018.

[33] A. S. B. Reddy and D. S. Juliet, "Transfer learning with resnet-50 for malaria cell-image classification," in *2019 International conference on communication and signal processing (ICCSP)*, pp. 0945–0949, IEEE, 2019.

[34] D. Srivastava, R. Wadhvani, and M. Gyanchandani, "A review: color feature extraction methods for content based image retrieval," *International Journal of Computational Engineering & Management*, vol. 18, no. 3, pp. 9–13, 2015.

[35] I. Kurniastuti, E. Yuliati, F. Yudianto, and T. Wulan, "Determination of hue saturation value (hsv) color feature in kidney histology image," in *Journal of Physics: Conference Series*, vol. 2157, p. 012020, IOP Publishing, 2022.

[36] H. F. Atlam, G. Attiya, and N. El-Fishawy, "Comparative study on cbir based on color feature," *International Journal of Computer Applications*, vol. 78, no. 16, 2013.

[37] U. Erkut, F. Bostancıoğlu, M. Erten, A. M. Özbayoğlu, and E. Solak, "Hsv color histogram based image retrieval with background elimination," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–5, IEEE, 2019.

[38] A. Nazir, R. Ashraf, T. Hamdani, and N. Ali, "Content based image retrieval system by using hsv color histogram, discrete wavelet transform and edge histogram descriptor," in *2018 international conference on computing, mathematics and engineering technologies (iCoMET)*, pp. 1–6, IEEE, 2018.

[39] C. Tomasi, "Histograms of oriented gradients," *Computer Vision Sampler*, vol. 1, pp. 1–6, 2012.

[40] M. G. Mohammed and A. I. Melhum, "Implementation of hog feature extraction with tuned parameters for human face detection," *International Journal of Machine Learning and Computing*, vol. 10, no. 5, pp. 654–661, 2020.

[41] J. Chen, D. Zhou, Y. Wang, H. Fu, and M. Wang, "Image feature extraction based on hog and its application to fault diagnosis for rotating machinery," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 6, pp. 3403–3412, 2018.

[42] G. H. Granlund, "In search of a general picture processing operator," *Computer Graphics and Image Processing*, vol. 8, no. 2, pp. 155–173, 1978.

[43] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[44] Y. Wicaksono, R. Wahono, and V. Suhartono, "Color and texture feature extraction using gabor filter-local binary patterns for image segmentation with fuzzy c-means," *Journal of Intelligent Systems*, vol. 1, no. 1, pp. 15–21, 2015.

[45] R. Hammouche, A. Attia, S. Akhrouf, and Z. Akhtar, "Gabor filter bank with deep autoencoder based face recognition system," *Expert Systems with Applications*, vol. 197, p. 116743, 2022.

[46] W. Xia, S. Yin, and P. Ouyang, "A high precision feature based on lbp and gabor theory for face recognition," *Sensors*, vol. 13, no. 4, pp. 4499–4513, 2013.

[47] S.-R. Zhou, J.-P. Yin, and J.-M. Zhang, "Local binary pattern (lbp) and local phase quantization (lbq) based on gabor filter for face representation," *Neurocomputing*, vol. 116, pp. 260–264, 2013.

[48] D. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun, "Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using deep learning algorithm," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pp. 966–971, IEEE, 2022.

[49] G. Zhao, Y. Liu, W. Zhang, and Y. Wang, "Tfidf based feature words extraction and topic modeling for short text," in *Proceedings of the 2018 2nd international conference on management engineering, software engineering and service sciences*, pp. 188–191, 2018.

[50] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[51] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, "Efficient agglomerative hierarchical clustering," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2785–2797, 2015.

[52] H. Li, K. Zhang, and T. Jiang, "Minimum entropy clustering and applications to gene expression analysis," in *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004.*, pp. 142–151, IEEE, 2004.

[53] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, *et al.*, "Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling," *arXiv preprint arXiv:2412.05271*, 2024.

[54] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[55] L. Chen, M. Zhai, J. He, and G. Mori, "Object grounding via iterative context reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[56] A. Franco, D. Maltoni, and S. Papi, "Grocery product detection and recognition," *Expert Systems with Applications*, vol. 81, pp. 163–176, 2017.

[57] B. Santra, A. K. Shaw, and D. P. Mukherjee, "Graph-based non-maximal suppression for detecting products on the rack," *Pattern Recognition Letters*, vol. 140, pp. 73–80, 2020.

[58] B. Santra, A. K. Shaw, and D. P. Mukherjee, "An end-to-end annotation-free machine vision system for detection of products on the rack," *Machine Vision and Applications*, vol. 32, no. 3, p. 56, 2021.

[59] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464–7475, 2023.

[60] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "Trocr: Transformer-based optical character recognition with pre-trained models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 13094–13102, 2023.

[61] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9865–9874, 2019.